AN ARGUS SPECIALIST PUBLICATION

October/November 1985

£1.95

ZX COMPUTING

The Magazine For All Sinclair Users



ZX goes to war with a look at strategy and war games

Go racing with the ZX81

QL Caverns Competition!

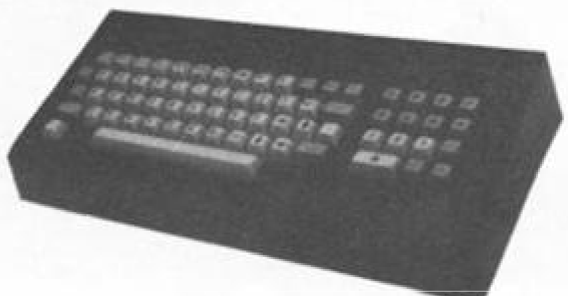
Microdrive Adventures

With More Editorial Pages Than Any Monthly Magazine

KEYBOARDS

SPECTRUM REPLACEMENT

THE 68FX2



ONLY **£39-95** INCL.

WE ARE NOW ABLE TO OFFER THIS TRULY SUPERIOR KEYBOARD AT THIS EXTRA SPECIAL PRICE **£39-95**
THIS KEYBOARD IS FANTASTIC
ITS FEATURES INCLUDE:

* FULL SIZE SPACE BAR
* INDIVIDUAL GOLD PLATED KEY-SWITCHES
* SINGLE ENTRY KEYS INCLUDE FULLSTOP, SHIFT LOCK, EXTENDED MODE, COMMA, FULLSTOP, EDIT DELETE BREAK, MINUS, *, RUN DIVIDE AND CURSOR CONTROLS
* 1/4 THICK PLASTIC CASING
* SIMPLE STEP BY STEP INSTRUCTIONS

ALL FOR THIS UNBELIEVABLE PRICE OF **39-95**

WE STILL HAVE SOME 68FX1 LEFT THESE ARE AS ABOVE BUT WITHOUT EXTENDED MODE, DIVIDE COMMA + MINUS ON SINGLE ENTRY KEYS ONLY

LIMITED STOCKS **£20-95** + 2-00 P&P

JOYSTICK INTERFACES

FOR THE SPECTRUM

THE ULTIMATE THE FOX PROGRAMME JOYSTICK INTERFACE:-



What the review said...
SINCLAIR USER FEBRUARY 1984
"Provides the easiest interface on the market to use"
SINCLAIR USER ANNUAL 1985
"At £28.95 the interface is excellent value"
ZX COMPUTING JANUARY 1985
"One of the easiest to use, get my approval"

THE FOX PROGRAMMABLE INTERFACE — AN INTERFACE WITH

* Complete compatibility with all programs.
* In-built memory to store up to 16 different games keysets at one time.
* Battery back-up so no loss of memory after power off. (Battery recharge during loss so no replacements are required).
* One switch only for simplicity of use.
* Full casing.
* Through port for further expansion.
* Proven compatibility with the microdrive.
* Built-in pseudo Rom facility with Rom for personnel tool kit.
* Compatible with all Atari-type joysticks.
* Compatible with Quickshot II rapid fire.

THE INTERFACE THAT ONCE PROGRAMMED NEVER NEEDS PROGRAMMING AGAIN!!!

NO TAPES!! NO LEADS TO CHANGE !!
JUST FLICK THE SWITCH AND PLAY THE GAME WITH FOX

SPECIAL OFFER ONLY!!
£28.95 INCL.

SPECIAL OFFER
BUY THE ABOVE INTERFACE WITH THE QUICKSHOT II JOYSTICK FOR ONLY
£34-95

OR BUY OUR STANDARD JOYSTICK INTERFACE

(KEMPSTON TYPE) ONLY £7-95 INCL

WITH QUICKSHOT II JOYSTICK £15- INCL

OR QUICKSHOT I JOYSTICK £13-75 INCL

SPECTRUM UPGRADE

16K TO 48K
(ISSUE 2 or 3 please state)

ONLY **£22-50 INCL.**

IN ONE SIMPLE KIT

SEND NOW TO
FOX ELECTRONICS LTD
FOX HOUSE
35 MARTHAM RD
HEMSBY
GREAT YARMOUTH
NORFOLK NR2 94NQ

PLEASE SEND ME:-

68FX2 KEYBOARD ☐
68FX1 KEYBOARD ☐
PROGRAMMABLE INTERFACE ☐
Q/S II + PROG. INTERFACE ☐
STANDARD INTERFACE ☐
STANDARD + Q/S II ☐
SPECTRUM UPGRADE ☐

Name ....
Address ....

I enclose £ ....
or Debit my Barclaycard/Access

"YOUR PROGRAMS ALWAYS REACH ME BEFORE ONES ORDERED LOCALLY"

(I. Sinclair, N.S.W., Australia)

WE ALWAYS TRY TO SEND YOUR PROGRAMS ON THE SAME DAY WE GET YOUR ORDER, THAT'S WHY IT'S CALLED SPEEDYSOFT! All programs normally in stock before we advertise them! FOREIGN ORDERS WELCOME, satisfied customers in 63 countries over 3 years. UK prices include VAT: export prices are the same, plus p&p because of the extra work involved. Pay by sterling cheque, bank draft or postal order. All orders must be prepaid: we send your programs as soon as we get your money!. It's fastest to order by phone from any country with your VISA, EUROCARD, MASTERCARD or ACCESS. Call 01-846 9353, 24 hrs, 7 days a week. BIG FREE ILLUSTRATED CATALOGUE WITH EVERY ORDER WITH DETAILED REVIEWS, LOAD TIMES, ETC. Send £1.00 cash (refundable) for catalogue only. FREE GIFT if your order from this ad comes to £30! See last paragraph.

EXPLODING FIST

"Karate should be a sure-fire winner when portrayed as realistically as in this superb simulation. The movement of the fighters is very realistic ... virtually beyond criticism, technically brilliant and visually absorbing." (PersCompWorld) "Each animated figure is a joy to see. Movement is fast, smooth and realistic ... just wonderful." (C&VG) For 1 or 2 players as they progress from novice to Tenth Dan. Over 700 sprites used for each fighter! 18 manoeuvres to learn. Instructions in French and German, too! Sticks OK. (Melbourne House) **Cassette £8.95**

RED ARROWS

"ONE OF THE BEST FLIGHT SIMULATORS TO BE RELEASED IN A LONG TIME." (HCW) "It's tremendously difficult to do and only absolute addicts should try it." (PCW) A faithful 3D graphic reconstrucion of the Red Arrows aerobatics as you train and eventually join the display team. Monthly competition to watch a real flypast ends 31.1.86! Sticks OK. (Database) **Cassette £8.95**

SOUTHERN BELLE

"PRETTY AMAZING ... possibly the most complex and realistic simulation I've yet seen." (HCW) "Train lovers will certainly enjoy a good, solid simulation and those who are used to flight simulations may find a day's outing on the Southern Belle a refreshing change." (SUsr) Complex steam-train driving simulation with easy training options to let you learn gently. Great graphics make you feel you really are on the footplate. No Stick. (Hewson) **Cassette £7.95**

Frank Bruno's BOXING

"In the battle of the boxers, it's a knockout to Frank ... stands head, shoulders and boxing gloves above the competition." (YrSp) "A great sports simulation. The animation is great and play action totally addictive. The graphics are some of the finest around for the Spectrum." (C&VG) Meet 8 successive heavyweights (each fighter loads separately as you qualify to fight him.) Each fights differently. Sticks OK. (Elite) **Cassette £6.95**

NATO ALERT

A wargame simulating a non-nuclear conflict in Europe between NATO and Warsaw Pact forces. 1 or 2 players, 5 skill levels. Up to 21 different objectives may be chosen, to determine length of play. Complex, horrifyingly realistic. Map graphics help. No Stick. (CCS) **Cassette £5.95**

NOW GAMES

6 famous programs on one tape for only £8.95! LORDS OF MIDNIGHT (Beyond) : BRIAN BLOODAXE (The Edge) : EVERYONE'S A WALLY (Mikrogen) : STRANGELOOP (Virgin) : FALCON PATROL 2 (Virgin) : ARABIAN NIGHTS (Interceptor). Save £39.00 on the original prices! (Virgin) Cassette £8.95

The ARTIST

"Who needs a Macintosh when you've got a Speccy and this program? It's features are powerful and easy to use. There are on-screen commands and a very fast and flexible FILL command and even a cut-and-paste facility. All it needs is a mouse and you've just saved yourself two grand ... **If I was asked to pin down the best program on its UDG handling alone, I'd plump for the Artist." (YrSp)** Claimed to be 'the most powerful and flexible graphics program available for the Spectrum.' Wafadrive/Microdrive/Discovery 1 compatible. SofTechnics will provide colour dumps of your pictures, too! Sticks OK. (SofTechnics) **Cassette £12.95**

FRANKIE/HOLLYWOOD

"A BEAUTIFUL PROGRAM, an adventure of great imagination, with some super programming techniques in the use of expanding windows and icon-like objects to pick up and use." (SUsr) "Somehow they have crammed a game of dazzling style and complexity into a micro. Playing is a true voyage of discovery. Relax! It's the most infuriating and addictive game this year." (PCW) Over 60 tasks in your journey from Mundanesville to the Pleasuredome: 23 clues needed to solve the murder: 8 arcade games to beat on the way! Sticks OK. (Ocean) **Cassette £9.95**

HOME PACK

This Home Software Bumper Pack from Gemini contains 4 of their most popular and most useful home business programs in one package on two cassettes with 4 helpful manuals. **DATABASE : GRAPH PLOT : HOME ACCOUNTS : MAILIST Fully Microdrive compatible. (Gemini) 2 Cassettes £24.95**

BORED OF THE RINGS

"An excellent and vast parody I would rather spend months unravelling 'Bored' than days on a self-important 'serious' epic ... Challenging and atmospherically written." (PCW) Huge, funny, beautifully graphical adventure spoof loads in 4 parts. No Stick. (Delta 4) **CASSETTE £6.95**

MICRODRIVER

PERFECT TAPE/MICRODRIVE TRANSFER AT THE PRESS OF A SINGLE BUTTON! *"Menu-driven, very easy to use and well error-trapped ... I much preferred the Microdriver." (YrSp)* Dumps the whole of memory to microdrive. Save and reload *any* program. No use to pirates as Microdriver must be present to run Saved programs. Specially designed ROM fits onto the back of your Interface 1 or 2. Through connector for add-ons. (Mirage) **£44.95** *including p&p*

MICRODRIVE CARTRIDGES

20 Sinclair cartridges in a stylish black plastic box (by Transform) with 20 individual compartments and room for index in the lid. Counts as one item for p&p. BOX ONLY for your own carts £5.95. **20 CARTs PLUS BOX £49.95.**

The FOURTH PROTOCOL

"The icon graphics are excellent and the speed is incredible ... one of the most enjoyable, complex but easy-to-operate games which I have played in a long time." (SUsr) There are three separate episodes to complete against a deadline in this exciting icon-driven adventure based on Frederick Forsyth's book. the state-of-the art for adventurers! (Hutchinson.)**CASSETTE £12.95**

NODES OF YESOD

"A SUPER GAME ... superbly drawn ... jumping is accomplished using a series of back-flips, which are beautifully animated ... great music at the beginning and some tolerable speech ... a real challenge ... somebody at Odin really took a pride in their work." (HCW) 256 beautiful and intriguing screens to conquer. Sticks OK. (Odin) **CASSETTE £9.95**

FREE TAPES M'CARTS

If your order from this ad comes to more than £30 *(excluding p&p)* we will send you 2 free 15min blank data tapes or 1 free microdrive cartridge. Indicate which on your order!

VISA ACCESS

SPEEDYSOFT

01-846 9353 (24 HRS)

EUROCARD MASTERCARD

POST TO: SPEEDYSOFT (ZX14)
37 CHURCH ROAD,LONDON SW13 9HQ, ENGLAND.
For CATALOGUE ONLY, send £1 cash. Refunded with your first order.
I own a SPECTRUM. I enclose a cheque/PO payable to SpeedySoft
OR charge my VISA/ACCESS/EUROCARD/MASTERCARD

No. ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Signature: ______ Expiry Date ______

Please write clearly. If we can't read it, you won't get it.

Name: ______

Address: ______

______ Postcode: ______

PHONE NO: if any, in case of query ______

Program Name	Price
Postage & Packing UK add 75p per order	
Europe ADD £1.00 per program	
Outside Europe ADD £1.50 per program	
Total Order	

ZX COM

Editor: Ray Elder
Editorial Assistant: Cliff Joseph
Group Editor: Wendy J Palmer
Software Assistant: John Gerard Donovan
Sales Executive: Alice Robertson
Advertisement Manager: Barry Bingham
Divisional Advertising Manager: Chris Northam
Production Controller: Sue Couchman
Publishing Director: Peter Welham

Origination and design by Argus Design Ltd., Circus House, 26 Little Portland Street, London W1N 5AF.

Published by Argus Specialist Publications Ltd, 1 Golden Square, London W1R 3AB.

CONTENTS



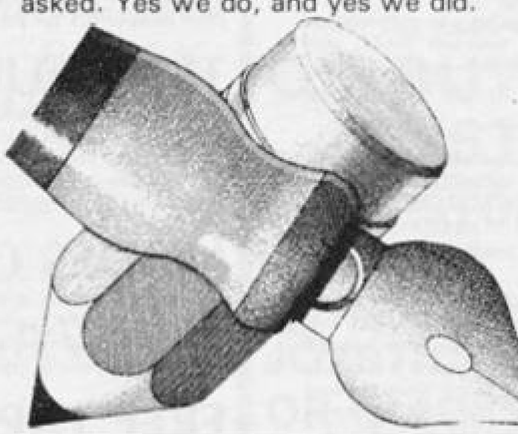
ASP FIGHTS SOFTWARE PIRACY

Much has been said and written in condemnation of software piracy but few have taken a positive stand against it. ASP is among those few that have taken action to help curb the grave problem of home copying of commercial software.

ASP has already taken steps to eliminate advertisements in our magazines which relate to tape duplication for piracy purposes. While it is appreciated that individuals may take 'back-up' copies of their own programs, it should be noted that it is *ILLEGAL* to copy commercially available software for other than personal use.

Software piracy is costing the software industry huge sums of money which is detrimental to the future development of the industry. It is in everybody's interests to dramatically reduce the level of software piracy primarily because firms need funds raised from software sales to plough back into research and development of new products. This means that the standard of software products can only improve.

ASP hopes our action will help combat this serious problem in order to maintain and improve the high standards of the UK software industry. We are asking *you* to do the same by refraining from duplicating or copying commercially available software for anything other than personal use.

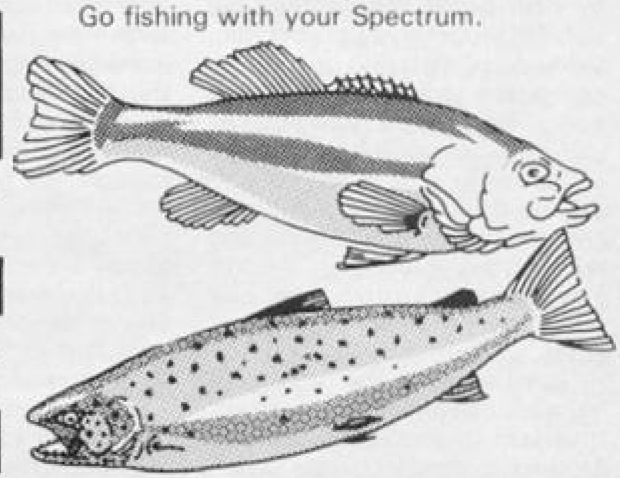




MPUTING

ZX Computing is published bi-monthly on the fourth Friday of the month. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed in the UK by: Garnett Print, Rotherham and London.







The summer holiday season draws to a close and I find myself dusting off the keyboard, TV, printer and cassette recorder, flipping the mains and settling carefully into my chair to begin writing this issue of ZXC.

I said carefully because I am in that state, well known to amateur sun worshippers, where the burnt skin still stings and the start of peeling has produced a terrible itching which is agony to scratch! I ought to be prepared for the British weather by now but it still catches me out, first four days pouring rain, followed by a blazing hot, sunny day which draws us all out of doors in pure ecstasy and makes us forget our good intentions to sunbathe in gentle sessions. I spent the next three days covered in various lotions and with no inclination to do anything much, except remember not to do it again next year!

But I enjoyed the break and have returned with renewed enthusiasm to tackle the tasks of producing the best mag possible.

Thanks

To everyone who asked after my health, I really appreciated it, especially as I did not expect any sympathy and only mentioned my visit to hospital to add weight to the Ethiopian appeal. Thanks also to all those who have bought the Soft-Aid tape, the whole affair is enough to restore even the biggest cynics' faith in human nature (Jonathon King excepted).

State of Play

Some time ago I mentioned that we had a fair amount of readers' programs left from the previous editor's reign and I promised that we'd get around to using them eventually.

Well, we are going through them slowly and I go through the whole lot when preparing each issue, so if you still have a program awaiting publication please continue to be patient, we'll get there in time.

Meanwhile . . .

We are still on the lookout for those extra special articles and programs that you've written.

To be honest, we have had several utilities submitted covering a wide range of topics but if you have written an unusual and ingenious program then we'd be interested in seeing it.

Again, we have a selection of the usual zap/chase/dodge arcade games for both the ZX81 and the Spectrum, most of which are very good. Originality is the key to getting a program of this type accepted, and if it includes machine code then we really are interested! So far though, we have had very few QL programs submitted. . .

Wargames and strategy programs are a little sparse. This is an area where we could do with one or two more programs (but *not* another "lemonade stall" or "Kingdoms" please).

Of course, ALL games submitted are looked at and considered, the time taken to reply to you may vary, usually it takes about two weeks to really assess your submissions but at certain times of the month, usually when copy is due at the typesetters, it may take a day or two longer.

Submissions

Sending in a program to us is really quite simple, all we ask are three things:

1. A typewritten explanation of what your program does, how it does it and operating instructions. Detailed descriptions of any ingenious routines are useful, as are line or section breakdowns of the program, and a variable list. However, these are *not* essential. All text should be typed and lines double spaced.
2. A tape with your program recorded on it, two or three times if possible, and checked that it loads back into your computer.
3. A listing of the program.

Items 1 and 2 are essential and item 3 is very useful to us, but we will consider a program if it arrives without a listing. If we decide that we cannot use your work then we will return all your material to you. If we accept your program or article then we will send notification and an offer in line with our current rates.

Finally I would just like to say tat although the standard of programming is very high, please don't be put off from sending in your work, if the idea is original and unique then we may make some suggestions as to how it can be modified with a view to using the end result!

Atic Atac

Dear Sir,
There are some things about 'Atic Atac' that I don't understand. I understand the obvious things, like a coloured key opening a door of the same colour, and that the spanner kills Frankenstein, but the things that puzzle me are; the bag of money, the leaf, skull, cross, and all the other things that can be picked up by pressing the 'Z' key. What are they for?
Yours faithfully,
Mark Thomas
Clwyd, Wales

I hate to admit it, but they puzzle me too. Are there any readers out there who can help with the solution? — Ed.

Sinclair Service

Dear Sir,
I am forever reading letters complaining about the after-sales service provided by Sinclair Research.

I recently had an annoying intermittent fault with ERASEing and VERIFYing my microdrive set up. I wrote to Sinclair Research in Camberley on the 9th June '85, explaining the problem and asking for advice on which unit to send for repair/replacement. A reply was sent on the 17th June, requesting the return of all my equipment. Everything went into the post on the 22nd June, and today (4th July) a parcel arrived containing replacement units.

The after-sales service has' proved to be excellent, and I think it's high time that Sinclair Research received more praise and fewer brickbats. Well done Uncle Clive (and his Camberley staff)!
Yours faithfully,
A. Jessop
Stanmore

Sinclair Research has come in for a bit of stick over its after-sales service on occasion, but in recent weeks they've adopted a new returns procedure, so maybe things will be improving — Ed.

Alien

Dear Sir/Madam,
I was one of the winners in your Unique competition, and would like to thank you for a copy of 'Alien' that you sent as a substitute prize.

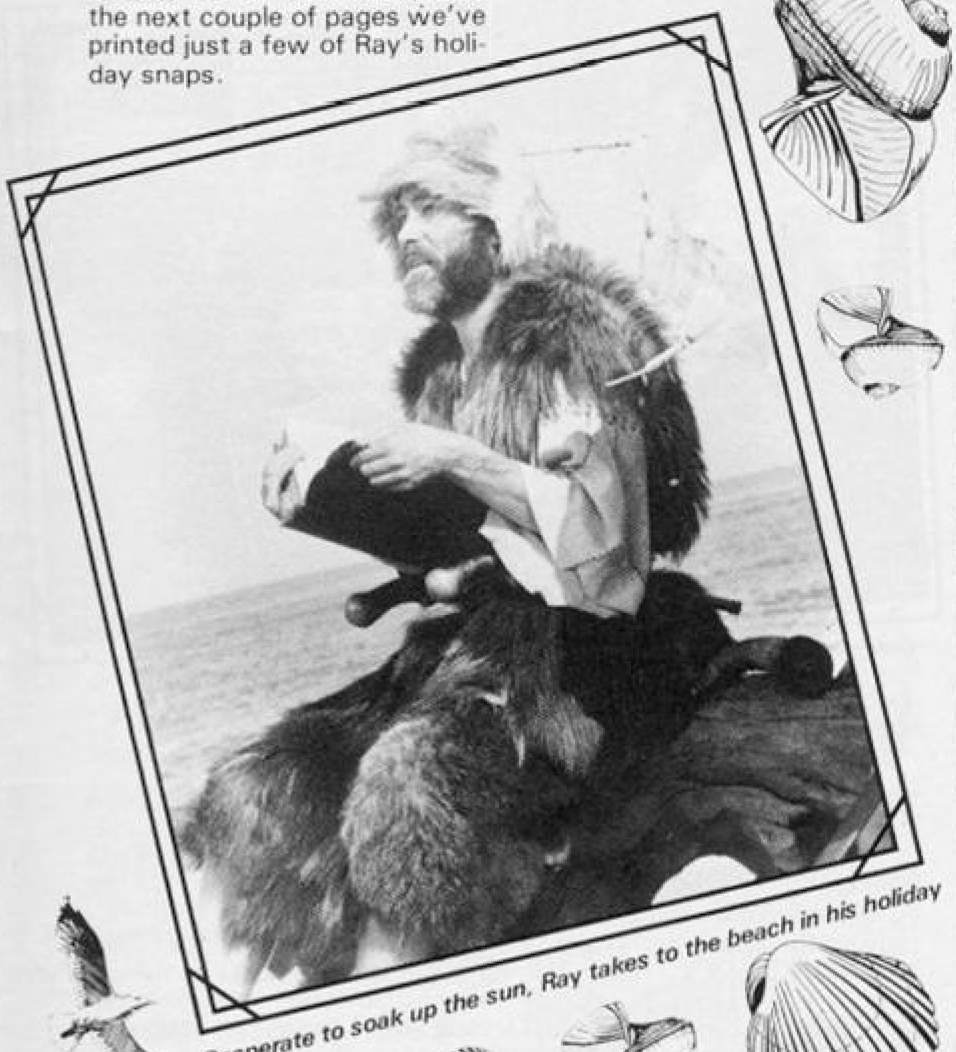
I would have written sooner, but unfortunately the tape arrived with no explanation, and it was only upon reading your June/July issue that I realised who had supplied the tape. Anyway, thanks for an excellent prize. As an avid arcade gamer I was extremely surprised to find Alien an extremely exciting and compulsive game, very much like the original film in fact.
Yours sincerely,
Leslie Grant
Hawick, Scotland

Glad you like Alien, and we hope that the other people that received copies were happy with it too — Ed.

Snap!

Now that Ray is back from his hols, I thought it would be nice to let you all see what our Esteemed Editor gets upto when

he lets his hair down. So, over the next couple of pages we've printed just a few of Ray's holiday snaps.



Desperate to soak up the sun, Ray takes to the beach in his holiday outfit.

Back issues and bugs

Dear Sirs,
I read with great interest the letters section of your magazine, and I am amazed that people expect every character to be perfect every issue. What could we complain about if every issue were perfect?

I am delighted with your magazine, and would like to take this opportunity to thank you. I am even more impressed by the recent response from you in sending me a new Feb/March issue, complete with corrections when my original subscription copy failed to arrive. Now I would like to ask — can I buy back issues back to Vol 1, No.1? If so, please advise me of the cost in US Dollars. If not, you need not reply.

Again, thanks for an excellent magazine — bugs and all!
Respectfully,
Dallas Swindal,
Hanahan, USA

Thanks for the compliment (it was a compliment, wasn't it?), however though we do normally stock back issues we have sold out of absolutely everything prior to Oct/November '84 — Ed.

ZX81 emulator?

Dear Sir,
At Christmas I bought tapes for my grandchildren's ZX81 computer, not knowing that their father had exchanged it for a Spectrum.

When buying a datacorder recently, I was told that there is now a tape available which, when entered into the Spectrum, will allow it to play the ZX81 programs. Try as I might, nobody else seems to know of this tape.

I have now been told that there was an issue of ZX Computing which discussed how this could be done. Please can you help?
Yours sincerely,
Mrs D. Davies
Swansea, W Glamorgan

The company that produced the tape you are looking for was called East London Robotics, but I'm afraid the tape only worked on early issue Spectrums, and is no longer on sale — Ed.

Great Spectextations

Dear ZX Computing,
I wonder if any other readers have experienced difficulties with the 'Spectext' program from Randle Hurley's book, 'The Spectrum Workshop'?

My BASIC program matches that in the book line-for-line (apart from an obvious error in the Save command which should read 'Line 6000, not 'Line 60000), and I entered the machine code according to the method outlined in the book.

Yet, when I attempt to RUN it, what should appear as spaces in the 'edit' mode, print as question marks ('?'). Following that, the program crashes with the message 'out of memory'.

I wrote to McGraw-Hill, the publishers, and after nearly a month received a letter saying that they'd passed my enquiry onto Randle Hurley, but, over two months later I am still waiting for an answer. . .

If anyone else has experienced (and possibly overcome) similar problems, I would dearly like to hear from them.
Julian Blackmore,
29, St Peters Close
Yaxham,
Dereham, Norfolk.

I'm afraid that none of the ZX team are familiar with Hurley's book, so we can't really help, but perhaps some of our other readers will be able to contact you — Ed.

INformation

Dear Ray,
You were kind enough to publish my letter concerning the oscillation of the values of 189/255 for the IN signal with no keys pressed, which I thought might have been caused by changing circuit values with temperature.

I have since had an experience, an account of which may be of help to other users of the Spectrum.

I use a W.H.Smith cassette recorder (type CP8300), fitted with a signal level indicator. For a short period this recorder would not SAVE correctly, so I carried out a few simple experiments, which I won't detail here, but the following details are important:

1. The signal indicated a reading as soon as the TV was turned on.

2. The reading rose to almost half way up the scale when I placed my hand between the television and recorder.

3. The reading decreased and the recorder would SAVE correctly when I moved it further away from the television.

4. With the 'mic' or 'ear' connections made, the 189/255 oscillation was present for the IN program.

5. With these connections removed, the computer returned a steady 189!

From these observations I concluded that there was a bad connection in the recorder which gave rise to a capacitive effect whose reactance at VHF was

Ray's holiday residence. . .



lower than the resistance of the circuit to normal signals, and so was picking up signals which were over-riding the signals from the computer.

(If there's anybody out there who understood all that, could you please let me know? — Ed's Asst.)

I returned the instrument to W.H.Smith, along with a report on my experiments. After one week only (excellent service) the recorder was returned to me with the spurious reading gone. It now operates perfectly. But:

a) The 189/255 oscillation still persists when the recording leads are plugged in, and settles at a steady 189 when they are removed.

b) The computer and T.V. therefore operate correctly on their own.

c) The 189/255 phenomenon is caused by the exterior additions made via the 'mic' and 'ear' sockets. Either exterior or interior to the computer, the fault is in this circuit.

Perhaps one of your brilliant correspondents with more sophisticated instruments and knowledge would like to continue the investigation.

Yours sincerely,
Tom Moran
Romford, Essex

By a happy coincidence, we received the following letter on the self same subject — Ed.

IN again

Dear Ray,

I remember a couple of letters complaining about difficulties with the IN function on issue 3 Spectrums. In the German magazine "Happy Computer" (Aug. 1985) I just found an article on that subject. I send you my partial translation, not knowing about any copyright prerequisites and not having tried it out myself yet. But I think it will be of great interest to all readers mentioned above.

'If reading the keyboard with IN (e.g. IN 63486 for keys 1 to 5) does not deliver a constant value (use test listing) but will jump instead between 255 and 191 you must not be afraid of having a defective Spectrum. But, on the other hand this Mic/Tape pin of the ULA (pin 28) which uses (value 64). While scanning the keyboard, the signal at the EAR socket will be tested too (see manual, page 117). without any signal there, HIGH or LOW status is not defined. On issue 2 Spectrums this seems to cause no problems. But on issue 3 and 4 the missing pull-up-resistance shows up as a factor of interference. A 47K resistor between pin 28 of the ULA and +5V will be sufficient to keep input without a tape signal on LOW.

Test Listing

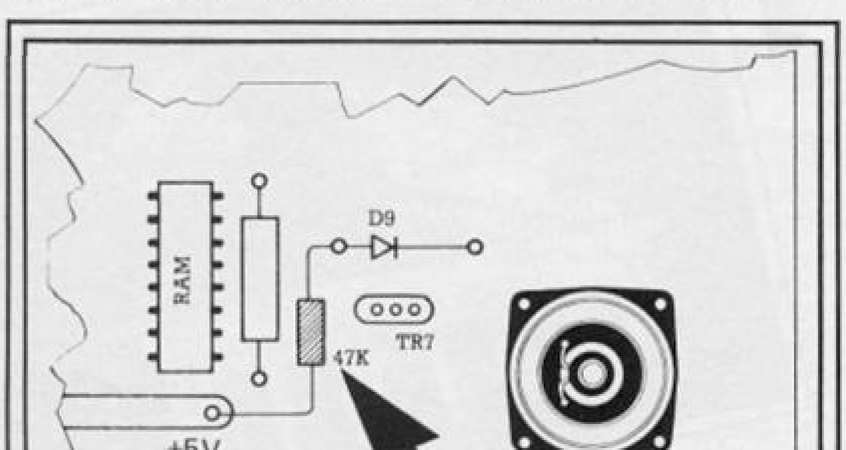
```
10 LET a = IN 63486
20 PRINT AT 10,10;a
30 GOTO 10
```

(by Jürgen Howaldt/mk)"

Louis Colombier
Marienburger Str. 17
5300 Bonn 1
West Germany

Hopefully that has explained the matter to everyone's satisfaction. Louis also sent along an illustration to further explain the point of this letter, and this is reproduced elsewhere on the page — Ed.

Circuit diagram — see Louis Colombier's letter 'IN again'.





Going out and about, Ray introduces himself to the locals.

Warts 'n all

Dear Editor,

I'd like to compliment E. Hutchinson for his article, 'Warts 'n All' in ZXC, Vol.2, No.7. We who are in the same situation as he, being completely self taught, will appreciate such help on the hard way from theory to practice. Also, reporting on previous mistakes and your present lack of full understanding can be a very good help for others. That is bravery!

We self-taught people may have this in common, that we have developed a fairly deep insight in limited fields, whilst we may still have vast gaps in others. And, that makes communication extra advantageous. So, please allow me to communicate a bit:

1) Since it has slipped the Editor's control, I think it should be mentioned that the limit between forwards and backwards jumps at JR orders is not 130, but 127. Anything above 127 will be understood as negative numbers. I think it is easy enough to remember if you think that from 128 onwards, bit 7 is set and can be understood as negative. *(Well of course we knew that, but we just wanted to make sure that you were all paying attention. That's our story, and we're sticking to it — Ed.)*

2) More about the JUMP orders; why should the value 2 sometimes be added when you are using a JR order? I think the following points may help one to remember why, once and for all.

When we are using any jump order in machine code, we are really manipulating the PC-register (program counter). This register always contains the address of the order to be executed next. Not just the next address, but the address of the next order, which must mean that it is able to recognise the different types of orders and remember how many bytes they use. We will come back to this.

As I mentioned, the PC register will contain the address of the next order to be executed. So, if you load the PC with 18000, then the machine will start executing orders at 18000. We could have used the mnemonics 'LD PC,18000', but we normally write 'JP 18000'. So far, so easy.

Now the relative jumps. Let us say that you are at address 15000, and you now want to jump to 15005 (relative). Since the JR order is two bytes long, the PC will already contain the number 15002. This must be changed. You add three to get 15005. We could have written 'ADD PC,3', but we have got used to writing 'JR 3'.

Now the relative jumps backwards. You are at 15000 and you want to go back to 14995. The PC already contains 15002, and this must be changed to 14995 by subtraction. You must subtract seven to reach that number. We could have chosen to write 'SUB PC,7' (or maybe 'ADD PC, −7') but we are used to expressing it as 'JR −7'.

3) At last — I always feel happy when I see a person use decimals for loading machine code. Why on earth bother to use hexadecimals? They only serve to make machine code look like such a highly mysterious science that no ordinary person will dare to attempt it.

What is really happening when you use a hexloader?

First you trouble your mind by looking up tables or counting on fingers to work out what, for example, decimal 10 would be in hex (which happens to be 0A).

Now, you want to POKE some address with that value, but the machine doesn't understand such a thing. So, you have to explain to it that you just want to express a quite ordinary decimal number. For that purpose, you must submit the '0A' to a longish treatment just to convert it back to the starting point (ie decimal 10). So why not tell the machine what you want from the start and use decimals?

Johannes Lind
Esbjerg, Denmark



In defence of hex

Dear Ray,
After reading the letter from Mr. Adams of Philadelphia in the Aug/Sept. issue of ZX Computing, I feel I must write to you in defence of hex.

Those of us who bombard you with machine code programs send hex listings not because we wish to show off an esoteric skill, but simply because when devising a machine code program, hex is easier to use.

To give one example, every time you need to enter a number larger than 255 into the memory, it must be entered in the form of low byte-high byte. The high byte of a number, *x*, is INT(*x*/256). The low byte is *x*–[256 * (INT(*x*/256))] Every time you need to store an address or any other number above 255, you must do those calculations. Get just one answer wrong and **CRASH.**

Take the number 12345. The high byte is 48 and the low byte 57. There is no way of finding that out other than doing the sums. The same number in hex is 3039h. The high byte is 30 and the low byte is 39 — no sums, you can see that just by looking at the number. This makes programming in hex much quicker and easier than programming in decimal. There are enough threats to your sanity involved in machine code programming without adding complicated arithmetic to your problems, I promise you.

However old fashioned Mr. Adams may think it, his computer is working in binary. However you feed information to a computer, be it BASIC, machine code, decimal, hex, Serbo- Croat or Mandarine Chinese, the first thing it does is convert it to binary so that it can make sense of it. We work in hex not to combat binary, as Mr. Adams suggests, but to work in harmony with it. It is no accident that the highest address in RAM on a 48K Spectrum is 65535. That is binary 11111111 11111111 — hex FFFF. It is the number of possible combinations of 1 and 0 in 16 bits. (16 — hex again. Hexadecimal means 16 digit arithmetic.) You might also call hex shorthand binary.

12345 in binary is 00110000 00111001. That does not seem obvious just by looking at the numbers. But take a look at the hex form:

Hex 3 = binary 0011
Hex 0 = binary 0000
Hex 9 = binary 1001

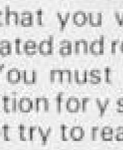
So, 3090h = 0011 0000 0011 1001.

Sorry, Mr. Adams, but until Sir Clive comes up with something other than binary — and I can't imagine what it would be — for his computers to work in, hex is here to stay.

Once you get to grips with it, and it is very easy to learn, you begin to wonder why anyone ever used decimal in the first place! Given a good hex loader, and there are plenty published, it is much easier to type a program in accurately in hex than to type a long list of decimal numbers in accurately. I think it is quicker too. After all, none of your numbers run to more than 4 digits and all the 3-figure numbers below 256d have only 2 digits in hex, so most of the time you have one less key to bash.

Perhaps *ZX Computing* could help those to whom hex is meaningless rubbish by printing a short course in hex arithmetic? In the meantime, there is a good short introduction in Toni Baker's 'Mastering machine code on your ZX Spectrum'. (Interface). Try it, Mr. Adams. If you could understand the structure of the program as you type it in, and that too is easier in hex than decimal, you might find yourself beginning to enjoy typing in listings.

Yours sincerely,
(Mrs) Carol Brooksbank
Coventry



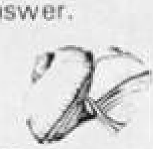
Decimal place

Dear ZX,
In the April/May problem page, in reply to Barry Murfett's question on how to make the Spectrum work to a fixed decimal place, you gave a rather long answer. I have a shorter routine which is as follows:

```
100 LET b=10↑m
110 LET a=INT(n*b+0.5)/b
120 PRINT a
```

In the routine 'n' is the number, 'm' is the number of decimal places one wishes to print, and 'a' is the required answer.

Yours sincerely,
Gautam Varma
Delhi, India



Rounding off

Dear David,
I recently received a complimentary copy of ZX Computing probably due to the fact that I market several T/S 2068 programs here in the states. I noticed in the issue that I have (April/May '85) that there was a question in your Sinclair Solutions column regarding having the Spectrum print with a fixed number of decimal places. (PRINT USING).

Several of my programs are laboratory based programs that required the ability to TRUNCATE and ROUND-OFF the tables of data generated. I ran across the following little routine that is very easy to incorporate into any BASIC program.

Before you PRINT your number (for example's sake, let your #=X) to the screen or printer add the following line:

```
LET X=INT ((X+.5/1Ey) *
1Ey)/1Ey
```

Where y= the number of decimal places that you would like X to be truncated and rounded off. NOTE: You must put a value in the equation for y so the computer will not try to read the E in the equation as a variable! Try this:

```
10 LET X=PI
20 PRINT X
30 LET X=INT ((X+.5/1E4)
   * 1E4)/1E4
40 PRINT X
```

You should get your final answer truncated and rounded off to 4 decimal places.

One final hint: if you have an extremely long or complex program that will require repeated use of the truncating feature, try this instead:

```
9999 DEF FN a(X) = INT
((x +.5/1Ey) * 1Ey)/1Ey:
RETURN
```

(Substitute y for # decimal places desired as in above example).

Now simply GOSUB9999 at the beginning of your program. When you are ready to use this feature to truncate and round off your value X use:

PRINT FNa(X) (instead of print X in your program).

As a final word I would like to say that I am extremely pleased with ZX Computing magazine, and will definitely subscribe to it. I would like to see more articles that cater to us TS 2068 users also. We are just now able to purchase the TIMEX of Portugal's 3" Disc drive.

Your sincerely
John Kuhn
North East Florida T/S Users Group
1707 King Street
Jacksonville
Florida 32204 USA

After a touch of sunburn, Ray goes undercover.



Shoptalk Shoptalk

Odds and ends, letters, and company info

Clean Machine

When you have been using your computer for a short while you begin to notice that it has a way of attracting dust and dirt.

Nu-Way Styli Components Ltd are marketing the products of Danish company AM Kemi, which consists of a range of cleaning aids such as the CKC-1 computer keyboard cleaners. This is a pack of 50 specially impregnated tissues which they claim will remove grease, dirt and smoke particles from your keyboard.

The PCC-5 kit is an interesting general purpose cleaning kit for screen, keyboard and 5.25" (or 3.5") disk drives.

Just for a change the company also markets an interesting looking joystick, I'll try and persuade them to send us some trial items and report back. I'm afraid they sent no information on prices or availability, so you'll have to keep your eyes open when visiting your local computer store.

Clean Machine II

I know cleanliness is next to godliness, but this is getting ridiculous.

Dennison Manufacturing Co, better known under their brand name for disks of Elephant Memory Systems, have produced four separate cleaning kits known as CK1, CK2. CK3 and CK4.

CK1 contains three different cans (or 'environmentally safe spray pumps') and 50 lint-free cleaning cloths. The cans contain a screen cleaner, a general purpose surface cleaner, and an antistatic solvent.

CKs two to four contain refills of each of the appropriate cans and 50 more cloths. Again, no details of price or availability were sent so the same comments apply to them as for Nu-Way. Actually, cynicism aside, this kind of product is very valuable as dust, dirt and static and a multitude of other ordinary household items can damage your computer. I have a friend who took his BBC to be repaired as it just refused to work one day. All they did was blow out the accumulated cat hairs from inside it.

Golden States

After a shakey start, the bullion seems to be flowing in from the US of A. When you think of it, the reason for the proliferation of programs on the US Gold label is that they have many of the major USA software houses under their banner and, as we all know, they do things in a big way in the states.

At the time of going to press EPYX are the latest of the US big names to join this label and in the very near future we will be seeing many, if not all, of their programs on our shop shelves. Titles that are likely to appear are; *Impossible Mission, Pitstop II, Breakdance, Summer Games II, Winter Games* and *FBI*.

Meanwhile, back at the ranch, US Gold and OCEAN have signed a deal to market products involving existing and future Walt Disney Characters. New films such as RETURN TO OZ, BLACK CAULDRON and old favourites like JUNGLE BOOK already have programs based on them under development.

Highly advertised and (consequently?) number one seller *Dambusters* now has its own Spectrum counterpart, check out our review of it.

Buck Rogers, a classic arcade zap-anything-that-moves game, also emerged from outer space and is a must for alien blasting freaks.

DATASOFT, another of US GOLD's contributors, finally produced *Pole Position* and are threatening us with *The Goonies*, the latest Steven Spielberg creation, and *Zorro*. It looks like it's going to be a hard year for us poor reviewers!

Elitism

Getting well immersed in the field of sporting activities are hyperactive Elite. *Grand National* was a respectable venture into the sporting world with a game which managed to combine strategy with arcade action and quality graphics. *911TS* is probably a weaker game and its main claim to fame is that it was produced under a promotion deal with SP Tyres and a free game was offered to purchasers of said tyres.

Their newest game really packed a punch and followed a well tried and tested trend of having a well known personality endorsing it. *Frank Bruno's Boxing* provides a painless way of enjoying the art of fisticuffs.

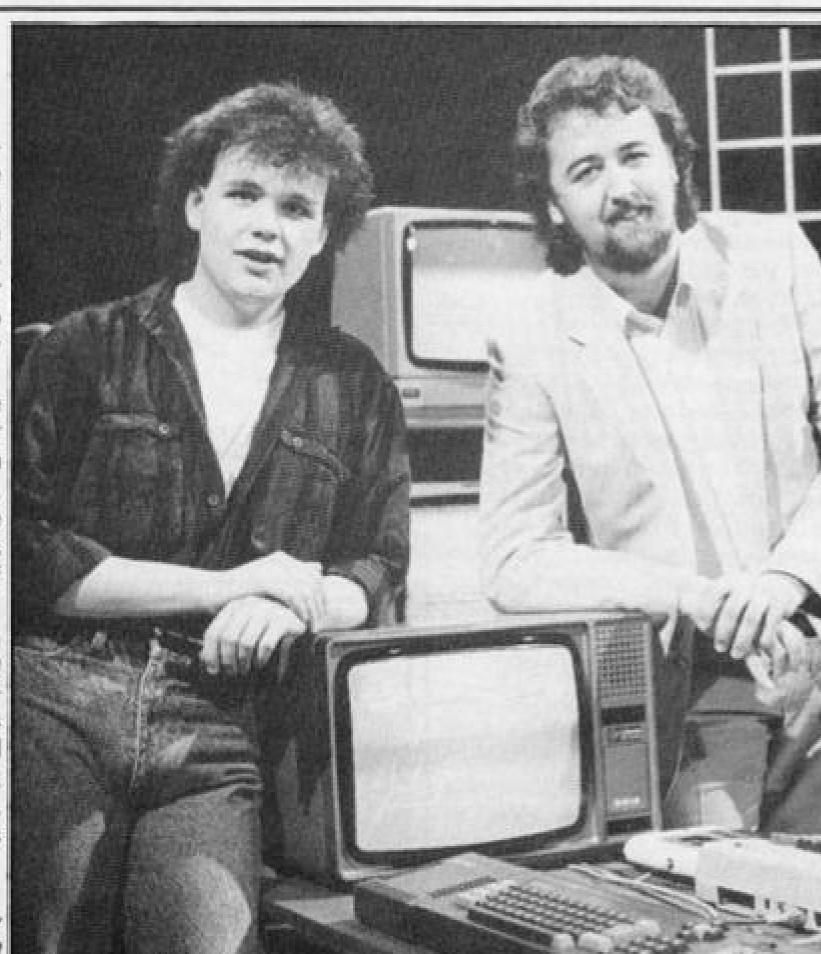
The latest of their offerings to entice you is *International Basketball* which they claim demands both skill and strategy

Grampian TV Bits'n'Pieces

No, not a comment on the quality of TV shows, but for all those lucky enough to be in an area where this show is networked, a mention that Grampian's second series of their computer show Bits'n'pieces is starting up any time now on a Saturday morning.

Men and machines are featured and Tony Crowther (*Monty Mole, Potty Pigeon*), Ian Bell and David Braben (*Elite*), Keith Campbell (adventure specialist), and Kevin Toms (Football Manager) are some of the former (I think).

Of the latter (the machines), most makes of computers are featured and speech, music synths, printers, modems, graphics and business will be looked at. I wonder why they forgot to ask me on for a guest appearance, perhaps they knew I was busy......



The picture shows presenter Bobby Hain (L) and the infamous Kevin Toms (R).

and this is likely to be followed by *International Soccer*. I must admit that I enjoyed *Boxing* and *G. National* and I look forward to seeing their forthcoming programs. By the time I've finished I'll have the fittest and most sporty fingers in the office!

The Best Days Of Your Life?

It's not often we're stuck for words, but the communication we had from St. Brides School (Burtonport, County Donegal, Ireland) has us scratching our heads bemusedly.

For a start the school is not a school, well not a real school as recognised by the department of Ed. & Science, but a sort of unusual holiday centre. Not that it's a holiday centre either. I'd better explain.

St. Brides provides 1920's style education, uniform, stockings etc. for girls over twenty. Nostalgia gone berserk! The whole atmosphere is related to the 'jolly hockey sticks' and period mannerisms, and, from their literature they seem to go way over the top. *The Secret Of St. Brides* (of course) is a Quilled adventure using The Illustrator for graphics and as such must be assessed in comparison with other Quilled games.

The game itself is difficult enough and the problems will take some unravelling, I must admit to not getting very far, but the 'Girls' Own' style of the language and the graphics, simple but effective, only served to irritate me.

The text is well written although there is evidence of lack of planning in that many words have to be split between one line and the next.

Honest opinion? Good, but expensive at £5.95. It has moderate appeal for adventure fans. As a first effort this is not bad, if they use the experience gained from it then subsequent offerings should be of a very high standard. The documentation, two professionally printed leaflets, is superb and puts many a professional to shame.



David Adams with his award for completing Technician Ted.



Debbie Sillitoe presenting Terry Jeffries, of TBD, with an award for sales of Dragontorc.

Hewson's Rogues Gallery

Lots of pics from Hewson Consultants this issue. First, holding up his silicon slice, is cheerful chappie David Adams. David is the first person to complete their game *Technician Ted* and was awarded the silicon souvenir by Hewson Consultants. For those still struggling a help sheet is available from them on request.

The happy twosome are Terry Jeffries of Terry Blood Distribution and Debbie Sillitoe of HC. She is giving him the clock as a token of gratitude for TBD selling the most Dragontorc programs. Now it is no secret in the office, or elsewhere for that matter, that I am a fan of Dragontorc, and the same techniques have been employed in the new games written by the fellow in the third photo. His name is Steve Turner and in his new game *Astro-Clone* he promises animation of the characters such as we've never seen before. It seems we've heard it all before, but HC has a habit of delivering the goods!



Steve Turner, author of Hewson Consultants' new game, Astro-Clone.

In . . . . Brief

• Owners of Microsphere's *Omnicalc* will be intereted in the *Omnicalc 2 Extension Kit* for £6.95.

This program has three main features, to allow you to interface with a Wafa or Disk drive, to enable single or double height screen dumps to be made on a graphics printer and a routine to print out all the equations used in every cell of your spreadsheet. Microsphere Computer Services Ltd. 72 Roseberry Rd. London N10 2LA.

In . . . . Brief

• Melbourne House has the backing of Jeoffrey Thompson, World Karate champion 82-84, for their latest offering, *The Way Of The Exploding Fist* which, unlike many similar programs that only have a few movement options, has an amazing eighteen possible player moves. £8.95 from most respectable take aways.

• For any Dentists among our readers, rush a cheque to Dentron Computers Ltd. 71 London Rd. London NW8 0DQ for £49.95 inc. VAT and P&P and they will send you *Charge Master*. This program will speed up and improve the accuracy of your patients' bills — and it won't hurt a bit.

• Activision will have shown their latest range of programs at the PCW show, just a bit too late to review for this issue. Games we will be looking out for on our visit will be *Great American Cross Country Road Race, Tour De France, World Champion Boxing, Great European Cross Country Road Race, Mind Shadow, Rescue On Fractalus* and *Hacker*.

• Look out for PSS's compilation tape '55' which contains their best five games so far.

• *Blast*! No, I'm not swearing, just giving you the title of Oxford Computer Systems' new compiler which they quote as being suitable for ANY Sinclair computer (ZX81 & QL?). Seems too good to be true, but if it is then it'll be worth every penny of the £24.95 asking price.

• Rod Cousens, late of Quicksilva, is now assisting Incentive Software on a consultancy basis for overseas promotions.

• Astrocalc, 67 Peascroft Rd. Hemel Hempstead, Herts. HP3 8ER have a range of Astrology programs if this is an area of interest for you. Including some for the XZ81.

• Still with the stars, Eclipse Software, 79 Ardrossan Gdns. Worcester Pk. Surrey KT4 7AX have improved their Astronomical program *Planetarium* and the second version is available for £6.95. Other titles include *The Cosmos, Meteor Showers, Messier List, Halley's Comet, Space Art, World Globe* and *Stargazer*.

• PSS should have released *Macadam Bumper* by now, this is a pinball game with a difference. The difference being that you can design your own tables.

There is also a competition to find the best designed pinball table using the program and the prize is a real pinball machine of your very own.

• Computer aided board games have been spectacularly unsuccessful, and we await with interest the latest venture into this field. *Bryan Robson's Super League* will cost £19.95 and can be played by two to eight players. I hope he has better judgement in his choice of software than with his England team.

• Quicksilva's *Glass* is a fast, furious and graphically stunning zap 'em all game. £7.99 at most local stockists. It'll make your joystick melt!

• Bad taste (but fun) award goes to 666, PO Box 190, Maidenhead, SL6 1YX for their game *Go To Hell*. It's a maze-chase type prog, but with the most Dante-like, gruesome and macabre graphics that I've seen on the Spectrum to date. Worth buying at £6.99.

• *Rocco* fell foul of copyright with its original name (*Rocky*). Produced by Spanish company Dinamic, the graphics are great but the action is limited, a bit like the old fashioned 'stand still and slog it out' method.

• A well deserved success came from relatively unknown company Bubble Bus with their *Wizard's Lair* program which took *Atic Atac* a step further. Their latest offering is *Star Quake* which they claim is even better than the last. We wait expectantly with £7.95 burning a hole in our pockets.

• An Interesting looking set of programs has emerged from Charles Letts & Co, Diary House, 77 Borough Rd, London SE1 1DW. Lett's *Keyfacts revision Software covers Biology, Chemistry, Computer Studies, Eng. Lit* — Henry IV pt1 and the Merchant of Venice, *Geography, Maths* and *Physics*.

Cost is £11.50 for each topic package.

• Buy one of the APS tape magazines, 16/48 Computing or Spectrum Computing and you'll also get the chance to win a personalised Austin Metro. Entry details on the October and November editions available at most computer software shops and newsagents.

• Latest from Imagine is *World Series Basketball*. At £7.95, it missed our publication deadline but we'll look out for it for next issue.

• CRL's much advertised *Rocky Horror Show* eventually arrived. We really liked it, it's a dodge and collect game with great graphics that recreate the original theatre sets. At £6.95 don't dream it. Buy it!

• *Abu Simbel Profanation* from Gremlin Graphics is a *JSW* variant which is devilishly difficult to play. If you completed *JSW* then give this game a go, it's a real challenge for expert arcade players.

• Sentient Software Ltd. should excite adventure fans everywhere, they have five interesting looking programs on the market and we rushed some to Brian Robb for his opinions for the next issue. Meanwhile the titles are *The Amulet, Scoop, Crystal Frogs, The Key To Time* and *Malice In Wonderland* and are £5.95 from Branch House, 18 Branch Rd, Armley, Leeds, W. Yorks. LS12 3AQ.

QL Software Scene

QL PASCAL

Metacomco has produced the first complete and approved PASCAL program for 68000 based computers to achieve full validation. There is an international standard for implementations of this language — ISO 7185 — and the British Standards Institution have presented Metacomco with a validation certificate.

QL PASCAL is compiled in a single pass generating true 68000 code which ensures fast execution times. Features include interfaces to QDOS, any length variable names, 32 bit integers, large sets and arrays and comprehensive error handling. Metacomco claim that because of the full implementation it is suitable for both beginners and experts alike. The language joins Metacomco's other programs for the QL, Assembler, BCPL, and LISP.

If you want to experiment with this language, then write a cheque for £89.95 and phone 0272 428781 to find out where to send it.

QL Games Deal

Microdeal Ltd of 41 Truro Rd. St. Austell, Cornwall PL25 5JE now have THREE games for the QL. Joining their first two, *Hopper* and *Cuthbert In Space* selling at £14.95 each is a staggering 2000 screen arcade adventure called *Lands Of Havoc*. This looks quite impressive from the screen pics sent, but it will cost you £19.95 plus 75p post and packing (on all programs).

The theme of the game is that you, Sador, have to slog it through the first 81 screens in

order to discover a book which will give you access to the lower levels and eventually to the Dark Lords who are mucking up this land of plenty.

Over one year in development this should be a good game, however we'll have to wait until next issue to get a full review.

A Talented Doctor?

The biggest criticisms of the QL tend to be aimed at the Microdrive storage system, so anything that can improve the reliability of the system is likely to be welcomed with open hands.

Talent Computer Systems (Curran Building, 101 St. James Rd. Glasgow, G4 0NS, who brought us some of the first games programs for the QL with *The Lost Kingdom Of Zkul* and *West*, each at £19.95 + 50p for p&p, have turned their hands to utilities.

The Cartridge Doctor may not improve the reliability of the microdrives but helps recover data and programs which would otherwise be lost entirely. Using techniques of artifical intelligence and incorporating a rule based expert system, the recovery process is almost entirely automatic. In some cases minor damage can be 'healed' by simply loading and resaving the information.

Files which are more badly damaged may not be so easily rescued but the program gives you the chance to inspect and patch the corrupted sections using a FILE PATCH utility. Other options include SALVAGE and TRANSLITERATE. The program comes complete with a comprehensive manual which Talent claims makes it easy for anyone to use — no knowledge of BASIC or machine code is needed.

At £21.95 it is a worthwhile investment for any owner of a QL who is using it for any serious, professional purpose.

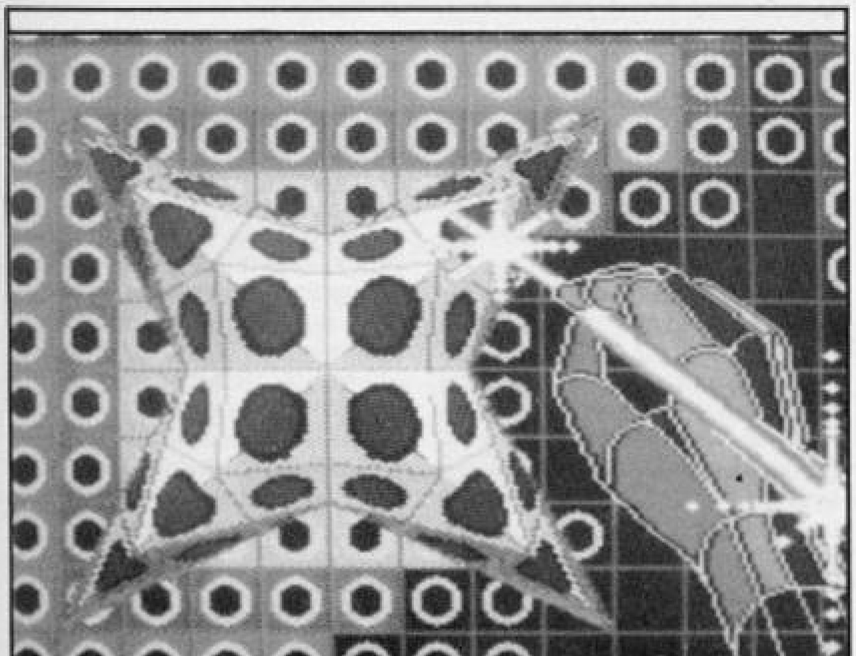


The Penman Plot

There I was, just settling down after a hard day's work and a hard boiled egg, when onto my TV screen burst Tomorrow's World and, before you could say Sinclair Research I was engrossed in the finals of the Prince of Wales Award.

And, who should be one of the finalists? Why, none other than our old friends Penman. Their plotting device will operate with any computer with an RS232 interface (as on Interface I) and provides versatile three (or more if changed manually) colour plotting facilities, and although it costs £249.00 it is by far the cheapest device to perform such a task.

Although it didn't win HRH's award it did win the Peripheral of the Year section in the British Micro Awards for 1985.



GraphiQL

Also from Talent is the most impressive graphics package we've seen for the QL.

We showed *GraphiQL* to Colin Christmas and he wanted to buy a QL to use this program on! For those of you who have been following his articles or those of you who are interested in graphics I'll give a quick breakdown of some of its abilities, freehand drawing in eight colours with optional flash, rubber banding including boxes, circles and ellipses, Doodle Pad, colour and texture fill, User definable brush (any width or colour), Air brush effect and many others.

The manual consists of 63 pages of text which is split into two sections, a tutorial and a reference section. The manual trots gently through the commands and provides a firm foundation on which to develop your artistic talent.

At £34.95 + 50p for p&p it is not cheap, but it offers the best facilities for CAD on the QL so far, and until someone else produces something else and proves it to be better (not an easy task) we have no hesitation in recommending it.



Spritely Logotron

Logotron Ltd. has developed an add on unit called The Spriteboard, which will provide you with the ability to create and control up to 32 sprites (sprites are moving graphics which pursue their own preprogrammed course until told to stop — usually when they collide with the operator's object.

The board will hopefully provide inbuilt commands for collision detection, shape and speed changes, shade and filling. There has been a version on the market for the BBC for some time now and the Spectrum version is due at any moment. The BBC board is priced at £179.95 but we are not certain of the Spectrum price. If you are interested write or phone them at Logotron Ltd, Ryman House, 59 Markham St, London SW3 4ND. (01-352-1088).

CGL In Control

Has the Joy gone out of your stick? Are those aliens laughing all the way as they zap you once again? Perhaps you could do with looking at CGL's range of Wico joysticks, most featuring sensitive lead switches and heavy duty sticks at a wide range of prices.

THE BOSS is priced at £14.00, The BAT HANDLE has dual buttons and costs around £25.00, The Three Way stick has three interchangable handles and is £27.00. The latest addition to their range is a budget priced joystick called the CHAMPION which costs a mere £11.00.

If you need an interface then they can supply you with one for £9.95. We cannot comment on their performance from personal experience as they haven't sent out any review samples, but we'll give them a phone call and see what happens.



In . . . . Brief

• Cheetah Marketing have moved to 1 Willowbrook Science Pk. Crickhowell Rd, St Mellons, Cardiff, Wales. This company has a wide range of useful peripherals and have several more planned for the Christmas period. One such item from the company (who once claimed 'The joystick is dead') is a new MACH 1 joystick!

• The power supply is the life blood of a computer system and in many areas it is none too reliable to say the least. Other problems may be trying to operate three or four units from one socket by the horribly dangerous means of bare wires. Nu-Way Styli products have a wide range of plugs, adaptors and connector boards at reasonable prices. For example, a Masterplug Powerclean, fused, would cost £18.99. Nu-Way Styli products are available in most electrical stores.

• Micronet seem to be gaining support, 6.5 million accesses in one summer month alone, and subscribers in excess of 17,000. I wish I'd bought those BT shares. . . .

• Interested in Data Communication? You could find Datalines of 68 Alston Drive, Bradwell Abbey, Milton Keynes, MK13 9HB (Tel 0908 311077) a useful company. Peter Woods tells us that they will send a free copy of their catalogue to anyone who writes or phones for one and this contains 48 pages of switches, cables, modems, multiplexers, terminals and testsets.

• Technology Research Ltd. has improved their already impressive disk drive interface and followed Sinclair and the BBC's lead in adding 'Plus' to its name.

The Beta Plus now features sequential and random filing capabilities, improved formatting system, a reset button and improved utility programs. A very useful feature is a RAM dump button which enables you make a copy on disk of any program.

TRL tell us that they will upgrade existing interfaces for £19.95 + £4.00 p&p and the new version will sell for around £95.00. Technology Research Ltd. are at Unit 18, Central Trading Estate, Staines, Middlesex TW18 4XE.

We'll have a close look at one for the next issue.

Hutchinson QL Suite

Five books have escaped from the Hutchinson Computer Publishing Company, each concentrating on a specific aspect of Sinclair's wonder machine, and each priced at £7.95.

Under the general title of The Sinclair QL Series, each is prefaced by the same introduction by the series editor, Robin Bradbeer, a well known computer author in his own right, and the same foreward by Nigel Searle of Sinclair Research. The topics themselves vary widely.

MACHINE CODE PROGRAMMING ON THE SINCLAIR QL is written by Martin Gandoff who, after a general introductory chapter jumps straight into number systems. Around chapter five we begin on 68008 machine code and assembly language. The style is an academic but friendly one, and for the serious programmer provides all the information to use the processor. I would not think that it would suit a novice wishing to learn machine code from scratch, but for those experienced in Z80 or 6502 etc. this is a definitive work on the 68008.

PROFITING FROM THE SINCLAIR QL by Barry Miles is intended for the person who is planning to buy or has just bought a QL, and specialises in financial and spreadsheet applications. The book covers a wide range of abilities but always stays within the scope of the novice. Topics covered range from setting up to, printers and file handling, just under 50% of the book is on various 'applications'. An excellent time if your prime interest is in these areas.

Mike O'Reilly wrote DATABASE MANAGEMENT ON THE etc. . . as a practical primer on using Archive. Above one third of the book deals with general ideas behind Database systems and the rest concentrates on using Archive. If this is your intended use of the QL then this is a book worth looking at.

MAKING THE MOST OF. . . . . . by Dick Meadows is an introduction to SuperBASIC and has been produced with both the newcomer and experienced user in mind. The book contains many programs of different lengths and the subject titles, Graphics, Graph Plotting, Sorting, Statistics, Science, Engineering and Solving Equations, reinforce the serious nature of the beast.

USING GRAPHICS ON. . . . was written by Garry Marshall again, and takes the QL seriously and deals with charts, graphs, line graphics, turtle and three dimensional graphics. Not a space invader in sight! Thirtyfive of the of the book's 199 pages deal with the use of Easel and the remaining examples use SuperBASIC. This book seems aimed at those interested in CAD, art and computer graphics and is well worth examining in more detail.

Interface 2

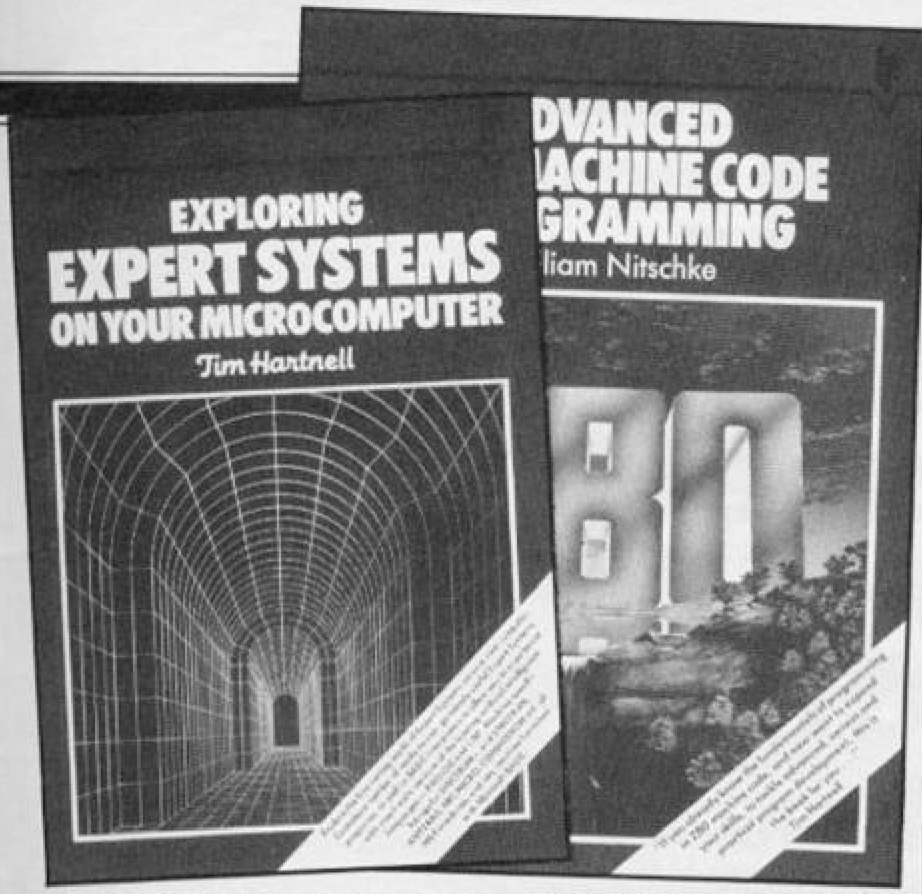
• For an issue to pass without Interface publications producing new books is almost unknown, and this grandaddy of computer publishers has just released two more.

Tim Hartnell, the original editor of ZXC, wrote the first one which is EXPLORING EXPERT SYSTEMS ON YOUR MICROCOMPUTER. This is an area which seems to have interested him recently and he is probably the home-user's leading authority. An expert System is basically a system which stores information and then by asking the user questions makes decisions. This book explains and discusses the topic in detail and provides several related listings in both machine specific and general form.

A fascinating book on a fascinating topic and priced at £7.95.

ADVANCED Z80 MACHINE CODE PROGRAMMING by William Nitsche is a book which is long overdue. There are many beginner's guides to machine code on the market and without exception (at least to my knowledge) they explain the language and how it operates but provide little if any information on how to actually USE this in your programming.

This book does not explain the 'grammer and spelling' of machine code but concentrates on program writing. Split into chapters titled Design, Structure, Messages, Organisation,



The Computer Game, The Business program, The Vicious Circle and Special Applications. This is an absolutely essential companion to your 'machine code for beginners' book, even at £12.95.

Elementary My Dear Watson!

I mentioned the Interface Advanced Z80 Machine Code Programming book elsewhere and said it was the perfect companion to a beginner's book and by sheer chance one such book came through the post. Published by Honeyfold Software Ltd, Standfast House, Bath Place, Barnet, London at £12.50, the package consists of a book and a tape.

The book is beautifully written in a straightforward way and my friend's fourteen year old son could understand it with no problem. It is very friendly and explains each instruction and section simply and clearly. There are lots of simple sample programs and each section is terminated with an exercise which helps you measure your understanding. The text print quality is a little poor but perfectly legible.

The tape contains a sophisticated Z80 assembler, Monitor and a Binary/Hex tutor program. This type of program often sells for the asking price on its own!

I was very impressed with this package and recommend it to anyone wishing to take the giant step into the world of machine code.

In . . . . Brief

• If you have a sneaking feeling that you could use a computer to help in your work and said work involves data, then DATA MANAGEMENT AT WORK by Kathy Lang may be the book you should buy. Essentially this book from Pan Books Ltd. is a discussion of the subject for those with no knowledge of the topic and helps you to identify important features in selecting suitable software.

• THE M&E EDUCATIONAL SOFTWARE DIRECTORY contains more than 100 entries listed under topics. I can't find a price for it, but every school should have one and it is available from Dept. BA1 MacDonald and Evans Ltd. Estover Rd. Plymouth, PL6 7PZ. Tel (0752 705251).

• Bernard Babani produce a superb range on a wide variety of subjects, their three latest are THE PRE-BASIC BOOK, a fairly painless introduction to computing (take note David Stuart!). INTO THE QL ARCHIVE, how to use the QL program for absolute beginners, and HOW TO WRITE ZX SPECTRUM, AND SPECTRUM+ GAMES PROGRAMS.

The latter was written by our old friend W. Simister and concentrates on board games rather than arcade games. At £2.95 for the first and £2.50 for the others these are good value for money.

Sinclair Saved?

Sinclair gets backing from creditors

In the last few days of August, Sinclair Research reached an agreement with its bankers and main creditors that will ensure continued financial and manufacturing support for the company.

Sinclair Research's debts are estimated at some £15m, owed to companies such as Timex, Thorn EMI, AB Electronics and both Barclays and Citibank.

As part of the deal, Sir Clive Sinclair will retain ownership of the company and a position on the management team, but will now concentrate on research and development. The new management team, will be headed by Bill Jeffrey, who has had some success with the marketing of the Sincalir Flat Screen Television. Precise details of the deal that has been reached are not available, but hopefully there will be more information coming in the next few weeks.



Sinclair renews QL push

In a further effort to stimulate sales of the QL, Sinclair Research has cut the price of the machine from £399.95 to £199.95. A spokesperson for the company commented that this major price cut was possible due to 'major savings from a substantial increase in manufacturing volume. . . we've also been able to cut costs through increases in efficiency based on experience gained in making QLs over the last year and a half.'

At the same time, Sinclair Research has announced its own 'approved' QL disk system manufactured by Microperipherals (full review next issue), and a range of new software titles for the machine, including QL Caverns, QL Monitor, Entrepreneur, Decision Maker, Touch 'n Go and Cash Trader.

All this, combined with the package deals for the Spectrum+ available from the retail chain Dixons, indicate that Sinclair are gearing up for an aggressive marketing effort this Christmas — especially since market figures now indicate that the Spectrum+ has a market share of some 35%.

If you've got millions of bits hanging out of the back of your Spectrum, you might be interested in the range of connector cables produced by Classified Products and Services.

Their Frank Purnell contacted us and sent us a smaple two-peripheral connector, which costs £12.50, but the company also produce three-way connectors and a range of cables for the QL. Further details from Classified Products on 0325-313131.

On-line with Prestel and Micronet

by Alan Giles

Prestel has always been a means for communicating with other people of like mind, whether in Closed User Groups or, if you want to talk to an individual whose name or number you know, you can just send a 'mailbox' message, without any charge apart from the normal Prestel and telephone usage charges. A message page is relatively tiny, compared with the amount of space on a written letter, but you soon get used to shortening the way you say thing to fit them on a single page, and you can always spread messages out over several pages if needed.

You can send the same message to a number of people, but this is quite time consuming, and a better idea is to make use of the various special interest clubs, like ClubSpot 810's Adventure Helpline, or Timefame 818's micro bulletin board.

One of the newest such clubs is the Micronet Gallery on page 323. This is a slightly different club, in that you essentially rent your own pages and can change their contents whenever you like, though your changes are only collected and actioned on a daily basis, unlike the alterations made by true Prestel Information Providers who can change pages directly themselves at any time they want.

The contents of the gallery pages are very varied — special interest pages on all sorts of topics — some amazing graphics considering the editing equipment that the designers are likely to be using — you name it, you could put pages up covering it (provided you didn't break the law or your Prestel contract).

If you want to put graphics on a Gallery page, you will need to be able to send the ESCape code CHR$ 27, a special telesoftware program in the HELP section of the Gallery adds a number of useful functions to your VTX5000, including redefining various keys to send a number of special codes. The Gallery help pages also describe the keys you need to press to get graphics, as does Appendix 1 of the Spectrum Micronet Book from Melbourne House. (*Which, by an amazing coincidence, Alan is the author of — Ed*).



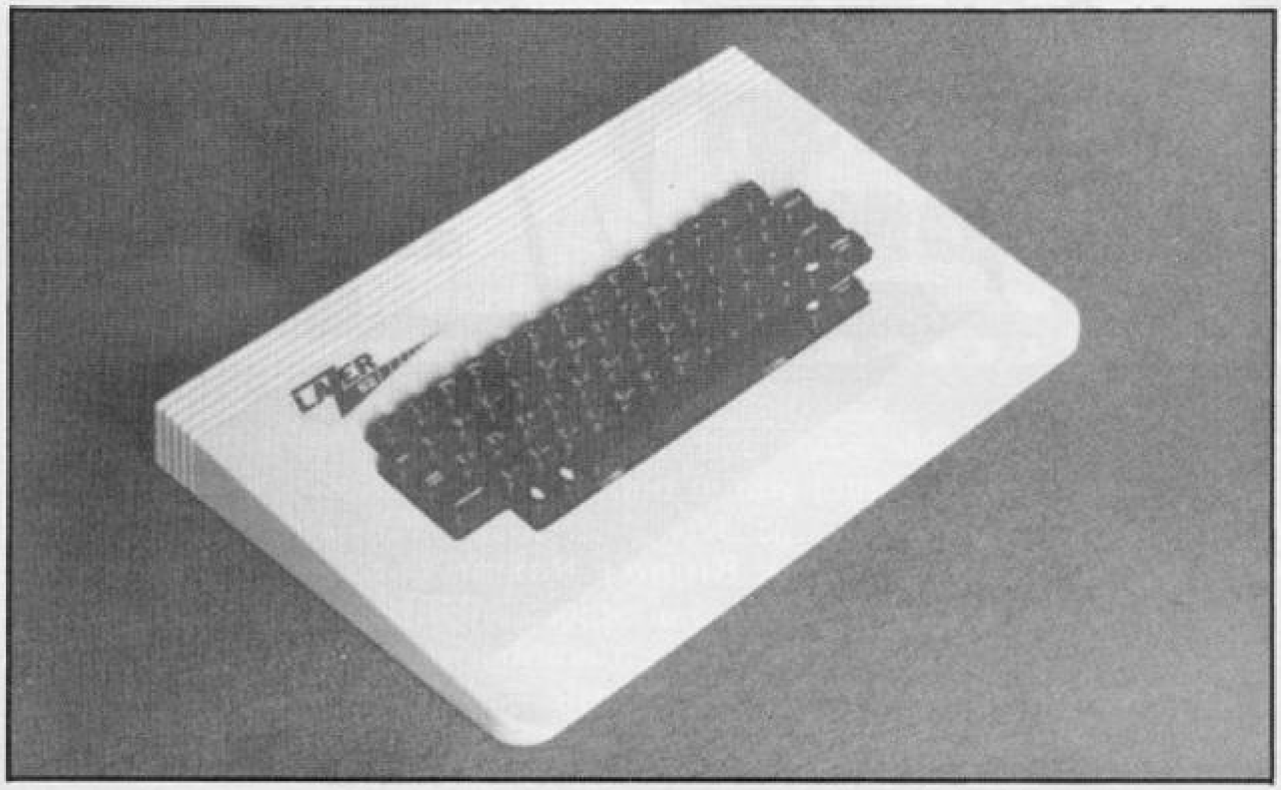
A bit of crackling . . .

One of the main problems with sending long messages over telephone lines is 'line noise' — crackles on the telephone line causing transmissions to be corrupted, resulting in spurious characters appearing on the screen. Unfortunately, when you see such spurious characters you cannot tell whether the Prestel computer has heard the noise in the same way as your VTX5000. The odds are that after line noise the cursor on your television screen is not in the same place as the Prestel computer 'thinks' it is — this means that there will be layout problems on the resulting Gallery page if you carry on without somehow realigning the two cursors. You cannot simply backspace (CHR$ 8), as both cursors will backspace and still be the same distance apart, and when you think you are writing over the top of the spurious characters and rubbing them out, you will probably be writing on top of something you wanted to keep. There are two ways of realigning the cursors — 'star zero zero' will clear and redisplay the whole screen and move both cursors back to the start of the input — this shows you how much effect the line noise has had on your input. Alternatively, the code for ENTER (CHR$ 13) will simply move both cursors back to the start of the input, sending this code is not achieved by pressing ENTER, as this sends the code for 'hash'. Either way you have the disadvantage that the cursor is back at the start, probably a long way from where you were typing, and during the time taken to key in the cursor movements necessary to get back to where you were, you may have yet more line noise, leaving you in a worse position than when you started. If line noise is particularly bad you may find it best to ignore it for a while, assume that the Prestel computer did not hear it (the modems at the Prestel computer areless susceptible to line noise than the VTX5000, so this is reasonably likely to be the case), do not backspace, just carry on as if there was no gap where the spurious characters have appeared. But, do remember to key 'star zero zero' before you SEND the message, to check how much effect the line noise has had. It is probably best to do this at the end of each character row affected by line noise.

Another way of reducing line noise problems is to reduce the time taken to transmit a message, by preparing it offline and sending it in one block. To help you in the preparation, you need to know that the Gallery message page has room for 597 characters, starting one space in on the first line — so it fills 15 lines apart from the first character on the first line and the last two on the last line. This is a lucky size, because the offline message size in the standard routines on the VTX5000 is 598 characters in b$(3 to 600), and you need to add a terminating 'hash' to a message, so the maximum VTX5000 message fits neatly in the available space. Remember also that when you use one of the ESCape colour codes it uses up two characters in the VTX5000 message string and in the available space in the Gallery message, but only one space on the screen. As the contents of b$ are manipulated in BASIC you should be able to see how to set up multiple pages and so on, further hints are given in the Spectrum Micronet Book. (You cannot use the VTX5000 routines without some change, as they insert ENTER and line feed codes in the message, using up valuable space).

Happy Communicating! ■

Laser 62 Keyboard



At one time the most important modification for a serious Spectrum user was to replace the rubber keyboard with a "real" one. With the development of the Spectrum+ this has not been such an essential requirement, although even that keyboard has its problems.

There are several keyboards available for the Spectrum covering a wide range of styles and offering a wide choice of extra features, so what does this one offer?

The keyboard is quite large, 15" x 10" which is as wide as a BBC and nearly as long. The thing that strikes you is the thinness, 1.5" at the back, sloping away to nothing at the front. It reminds me of the BBC in the finish, cream plastic case and dark grey keys. There are 62 keys, the main selection of extra keys is bunched at the right of the keyboard and consists of '$ #( )" – + :: = ,./ ✱' plus delete, E mode and up/down arrow keys. On the left are additional Edit, Caps Lock and left/right arrow keys. A full size space bar is fitted.

Key legends are of the stick-on type and are all printed in white, the extra keys are printed in large bold form and stand out from the rest. Generally a very smart looking product.

Fitting

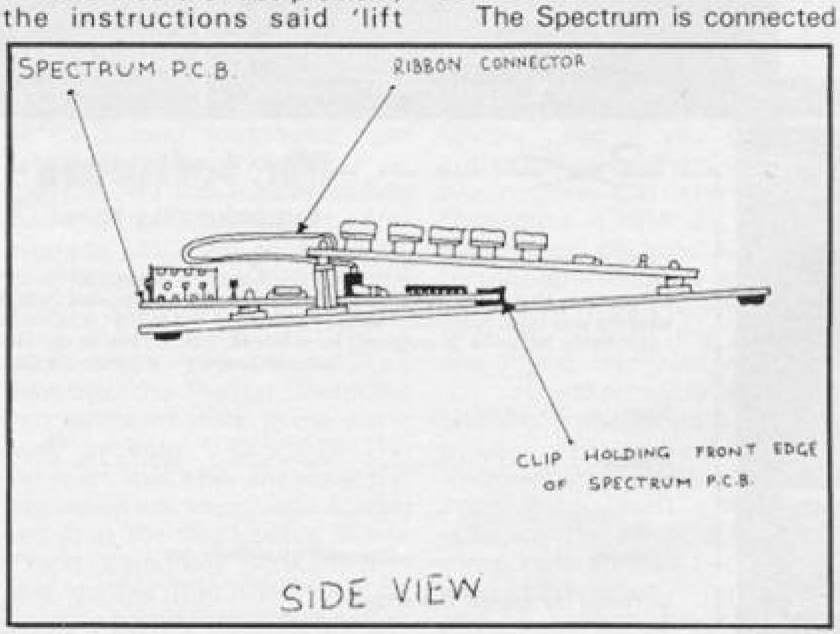
Whenever I have to add such a unit to a Spectrum I try to put myself in the place of an inexperienced owner and act dumb (*I'm not saying anything – Cliff*), and as such I think the instructions supplied would cause some concern.

The instructions are a photocopy of some typed notes accompanied by a simple, hand drawn, side view sketch of the assembled unit. Actually it is very simple to fit, and the system of fixing the PCB and key unit is unusual, this consists of a metallic baseboard with plastic poles stuck to it. The screw holes of the PCB fit over them and are held in place by a spring device at the top of them.

Here was the first problem, the instructions said 'lift keyboard from plastic mountings and fit the Spectrum PCB'. By pulling on one of the mountings I managed to break it. It was only by careful examination of the bits that I realised that the spring bit at the top had to be compressed to allow the hole to move over it. Silly me!

Slotting the PCB into position was no problem and neither was replacing the key unit although the fit is extremely tight due to the height of the casing. Screwing the bottom of the case to the top was OK, although there were more screws than I needed, and an interesting looking bit of metal and a rubber grommet remained without any explanation.

The Spectrum is connected to an On/Off switch by a flylead and the PSU lead is plugged into a socket supplied on the back of the case.



In use

First of all there is a problem for Interface 1 owners. There is no way the interface unit can be fitted inside the case and, although it could be forced to fit uncomfortably underneath, the only way of satisfactorily using it was to use an extension ribbon cable.

Secondly, because of the depth of the unit it would not fit comfortably on my desk with all the other bits and pieces. A massive rearrangement of the whole system was essential.

When I actually got round to using it, the white legends did not cause the confusion I had expected, this was mainly due to the choice, position, and bold labelling of the additional keys which seen to be the ones most frequently used.

The case is well designed and is comfortable to use, the bevelled front makes it convenient to rest your wrists on and the keys, though not stepped, are spaced and separated enough to make fairly fast and accurate typing possible. Unfortunately I did not like the plastic feel of the keys, and the space bar on the one supplied has a 'soggy' feel and kept sticking down.

But . . .

I wonder who will buy this keyboard. In its favour are the excellent extra keys, but against it is the fact that there are many others of similar, or better quality at about the same price. At £55.00 it is not cheap, and it is not a truly 'professional' keyboard for practical usage. I typed this article on it using Tasword II and found it reasonable but I preferred other boards that I'd used. For programming it has many advantages over the old Spectrum but is on a par with the Spectrum+ keyboard available at £20.00. Possibly I could see a use in Educational establishments where an all round sturdy unit like this could be a great advantage. I do think though, that it is a little over priced.

Micro Board International, The School House, Station Approach, Woking, GU22 7UY. £55.00 ■

POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE
FOR ZX81, T/S1000 and T/S1500 COMPUTERS

ZX-TEXT



A word processor is to a computer user what a typewriter is to a typist, except that the former has more advantages than the latter. ZX-Text can operate in 16-64K RAM providing from 1350 to 9000 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back-space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer. (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends...etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$11.95

ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters / spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters / cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX-81 / TS1000. It offers a unique *SUM function that totals one or more rows / columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows / colums. Loading of data into more than one cell can occur across / down one or more row / column simultaneously. With vertical windowing you can arrange a set of columns in any order, or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter / erase, move, calculate, print, save and clear the spreadsheet. Enter / erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire sreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute / relative replication, down / across a column / row, is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX / TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial model rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$11.95

ZX-CALENDAR



Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 30 appointments in 16K, 100 in 32K, 180 in 48K or 250 in 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search/check/sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently originate, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know, by using it, to never be caught astray by any time-frame.

$11.95

$1.50 SHIPPING AND HANDLING/PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** FLORIDA RESIDENTS INCLUDE 5% Sale Tax

ZX COMPUTING's REVIEW of Leonardo

I've had to wait a while for my copy of Leonardo so let me record a vote of thanks right away to Creative Sparks for their prompt response to our dear Editor's gentle hint to that effect. Expectations grow with waiting and the usual preliminaries are likely to go by the wayside once you get a chance to get to grips with the thing you have been waiting for.

So, without, as they say further ado, let's see what's offered by this Graphic Utility for the 48K Spectrum, Its claims are considerable, as one might expect from a package bearing the name of someone who might easily be described as one of the most creative sparks of the Italian Renaissance — 'simply the most advanced Spectrum graphics package. . . . . . . in fact the only restriction is your imagination.'

It certainly made an impression on my son. I had to pull rank to use the program myself, but not before he'd had time to produce at least one effective, if simple screen within a very short space of time. I include it if only to show that a quick start can be made with this package.

Keyboard and Joystick control are available and a 98 page manual is waiting to take you through the facilities offered. It's a comprehensive manual and worth noting here that it is written in a very effective and practical style. More like a guide at first, to some pretty heavy stuff which if approached in any other way would have almost certainly put off many a potential user.

The main Menu offers three Options; CREATE GRAPHICS, SAVE GRAPHICS and LOAD GRAPHICS. Starting to draw using the Pixel Cursor, a small flashing black dot on a white screen, is straightforward. This is the pen position which of course can be moved as if the pen were on the paper, or lifted in order to be moved onto another part of the screen. The current mode and the cursor position (measured in pixel units) can be called and displayed in the top left corner of the screen. Brush size — the number of pixels in a stroke - can be changed easily. After some experimentation this facility becomes quite easy to use.

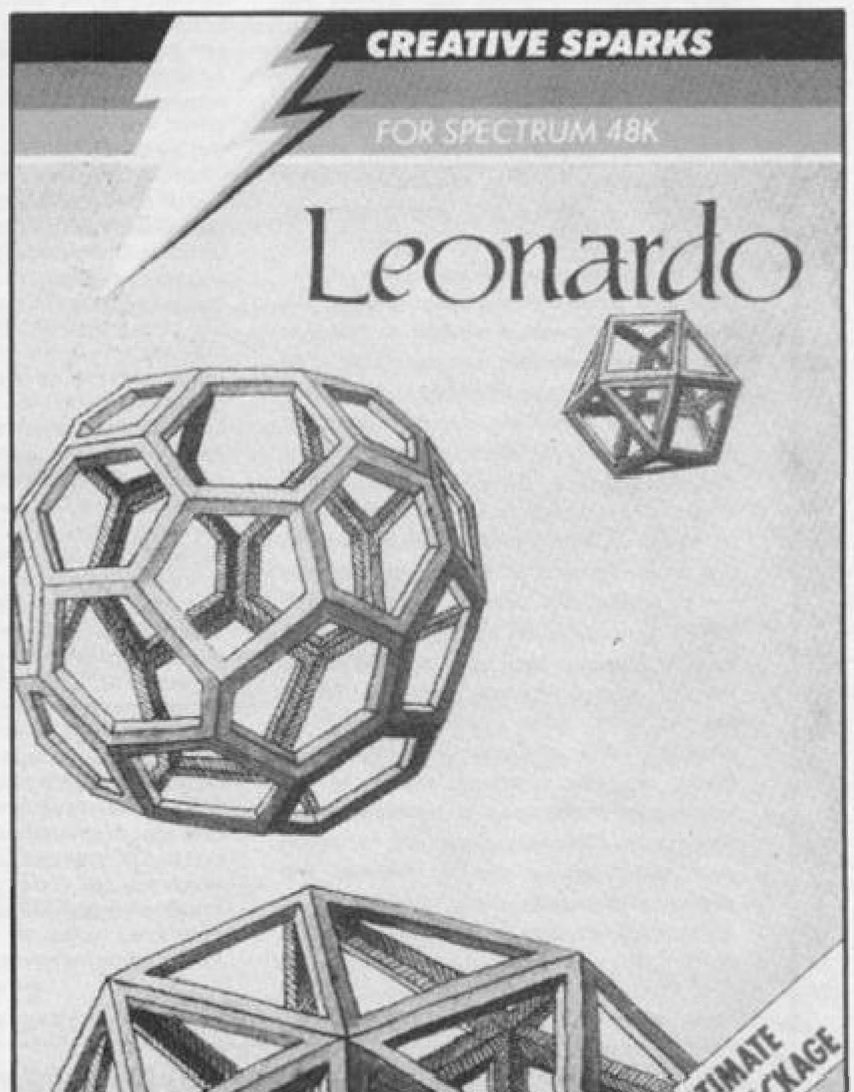
At any time the INFORMATION WINDOW can be called. This gives the user an update on the current state of the program. e.g. Mode, Cursor information, brush size, line information and so on. Curved lines can be drawn allowing some fascinating experiments with ellipses. At any time, those inevitable mistakes can easily be removed or erased from your drawing. There are no problems either adding colour or filling areas indicated by the cursor. The WINDOW facility also allows a great deal of scope for creating colour patterns on the screen. This same facility enables a small rectangular window to be defined in your picture.

Creating colour patterns is just one way of using the window, magnifying is another. Only a small area of the screen can be tackled at a time and the magnified image printed elsewhere. Words and UDGs can be incorporated into pictures and the 'blocks' so created can then be inverted or rotated.

Another feature offered by Leonardo is PROGRAMMED DRAWING, a way of creating and then drawing and repeating complex patterns and shapes. This facility is useful, but requires quite a lot of work and time spent with the manual.

Finally, scrolling can be achieved. Either the whole picture or only part within a window can be scrolled in any direction. Using the scrolling technique further textured patterns can be created.

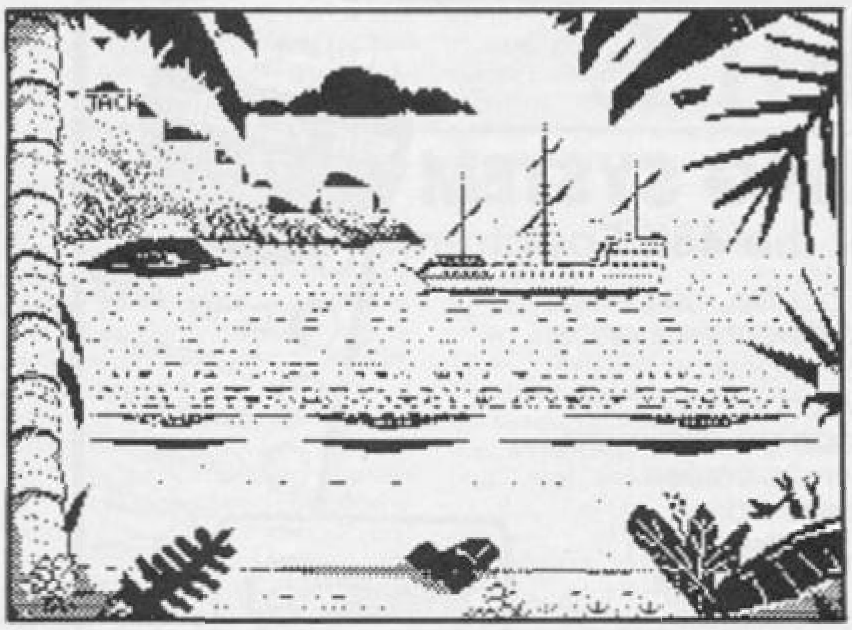
The whole program has enough prompts and signals for the user to have all the information he or she needs at any given time. Having said all this, and with so many Graphics packages now available for Spectrum users, some perspective is needed in order to assess utilities like Leonardo. I think it may well be time to do so. And I hope to be able to this next issue with special reference to Leonardo.



The Artist

Recently released in competition with 'Leonardo' is Softek's 'The Artist'. Carol Brooksbank tried it out for ZX.





The Artist
Softek
£12.95

My artistic talents put me firmly in the 'If-You-Lend-Me-a-Ruler-I-Can-Draw-a-Straight-Line' School of Art, but even I could produce a creditable screen after only a short acquaintance with this superb program. Anyone with real ability could produce spectacular work with it, as the examples on the tape show, and the graphics handling alone makes it worth the £12.95 Softek ask for it.

On LOADing, you are faced with a blank screen, two cursors, and a menu at the bottom of the screen. There are two other menus, and you can flip between them at will. One cursor is the reference cursor, which marks the centre of circles, the start of a line, etc. and the other is the operative one, which you move with the cursor keys. My only quarrel with the program is that three are Q, R, S and T. The handbook says that this is because they make it possible to use fingers to move the cursor and thumbs to change from 'ink' to 'rubber'. That's a nice theory, but most of us are brainwashed into thinking that you use arrow keys for moving things around, and I have lost count of the number of unwanted circles and squares I have produced by trying to move the cursor with keys 5 — 8. The program makes it easy to rub such things out, but it is irritating. The cursor moves very slowly when held down for short periods, making precise work possible, but if you hold the keys down continuously it speeds up, giving very rapid movement about the screen. There are three cursor modes. One leaves the reference cursor where it is until you move it, the second moves it up to the current position after each line is drawn, and the third makes the cursors follow each other around, so that you can draw parallel lines, or circles and squares of the same size. You can also draw 'freehand', using the operative cursor as a brush.

There are eight brush sizes, from one pixel to one character square, and also an oblique one, which is rather like writing with an italic pen. In addition, you have an 'air brush' mode for adding stippled shading. The brush can paint with the ink colour, or with one of several shading patterns. The same shading patterns are available for filling shapes, and you can also design your own patterns for this.

The fill operation is faultless. I gave it all manner of unlikely shapes to fill with all sorts of shading patterns, and it was perfect every time. You must make sure that the shape you wish to fill is completely enclosed, though. Even a one pixel break in the perimeter will let the shading bleed into the adjoining area. Fortunately, the program shows you the area to be filled before anything permanent is done, so you can trace and seal any gap. The enlarge mode is very useful here, if you are trying to find a one pixel gap.

You can at any point enlarge the drawing to allow you to work with precision on fine details. The window onto the screen moves about as you move the cursor, and you can switch between enlarged and normal size freely. At any time, you can overlay a chequer pattern which shows you the boundaries of the attribute squares, and you can set the attributes one square at a time as you work, or over a larger area if you wish. If you need to work on the bottom two lines of the screen, normally hidden by the menu, you can move the screen up.

There is a cut and paste mode, which allows you to move sections of your work about, copy a section to another part of the screen, or enlarge or reduce portions in either or both directions. If you are producing a symmetrical design, you need only draw half, because the cut and paste mode can produce a mirror copy.

At any point in your work, you can store the current state of the screen in memory, and if you then make a mistake you can return to the stored state. I found it useful to supplement this with the occasional tape saving, however, as the stored memory is automatically updated after cut and paste or fill.

Finally, there is a text mode with a choice of typefaces, including Gothic, and a very good small type face which is clearly readable even in a screen dump.

The program's UDG handling is superb. There are eight sets of graphics defined in the program, and you can redefine seven of them, the eighth being the Sinclair character set. You have





to be careful which you redefine, because all the shading patterns and type faces are held in the graphics sets. The program itself makes use of the small type face, and if you redefine that the menus start to look a bit odd.

The graphic defining grid allows you to define a figure which covers up to nine character squares, and which can then be manipulated. You define on an enlarged grid, but there is a normal size box alongside, so that you can judge the finished effect. You can save the finished figure in memory and print it to the screen. You can also pick up figures already on the screen if you wish to modify them. This means that you can, if you wish, define figures even larger than nine squares. You draw them on the screen as part of your overall design, and then pick them up from the screen in blocks of nine squares and store them in the graphics memory and/or on tape, microdrive or wafa.

If you are intending to animate a graphic you can define up to four related figures, and the program will show you the animation effect, by printing them rapidly one after the other in the box beside the grid.

The graphics mode can also be used to define your own shading patterns or character sets, and the program comes with a set of useful characters, including an aeroplane, helicopter, and tank, and a man with a gun who can be animated so that he runs along shooting.

All the character sets and your screens can be saved on tape. The program is microdrive compatible, and claims to be Wafadrive compatible, though I found that you had to tinker with the Basic a bit before you could save and load freely to Wafadrive. There is a compressor program on the tape, allowing you to compress all, or part, of your screens, store and recall them by number when required. Screen dumps to ZX printers are provided for, and I found that a very minor alteration to the Basic allowed me to produce screen dumps with my Epson printer and Kempston interface. The range of shading available in the program makes it particularly useful for producing small line-drawing illustrations on the printer.

The publishers invite you to register with them as an owner of 'The Artist', so that you will be entitled to discounted copies of future enhanced versions, and they offer a printing service. Send them a tape of your screen and they will provide colour printouts at prices ranging from 75p for a one-off 3" wide to £2 for 8" wide. Multiple copies are a bit cheaper.

I would recommend 'The Artist' to anyone who wants to produce artwork with the Spectrum, amateur or professional. However, good (or bad) you are as an artist, this program will help you to produce very elaborate screens better and more quickly than you thought possible. It is not often that you come across a program which so splendidly lives up to the claims the publishers make for it.

Carol Brooksbank

Across the Pond

by Mark L. Fendrick

Usually when a computer is no longer in production, the books which are written and published for that computer no longer appear. But, that hasn't proved to be the case with the Timex/Sinclair computers. In fact, the books which are now on the shelves in North America fall into two different categories. The first category contains those books which were written in the United Kingdom and imported to the United States. These books were written for the Sinclair ZX-81, ZX Spectrum, and the QL. In the case of the ZX81 books, they are completely compatible with the T/S 1000/1500 and can be utilized with no problem at all. The books written for the ZX Spectrum are mostly compatible with the T/S 2068, but since the T/S 2068 has a modified Spectrum ROM which includes additional features, they are not included in these British books. These imports, however, made up the bulk of the early offerings for the T/S 2068, often with their titles altered to *appear* to be written for the T/S 2068.

Now, however, books written specifically for the T/S2068 have been written and published. It is a few of these books that we will consider this time.

T/S Books

We'll start with a look at a book entitled; TIMEX SINCLAIR 2068 BEGINNER/INTERMEDIATE GUIDE by Fred Blechman (published by Howard W. Sams & Co., Inc; 4300 West 62nd Street; Indianapolis, Indiana 46268). A long time fan of the Sinclair computer, Fred is a believer in teaching practical applications rather than just theory, so you will find clear plain English descriptions with programs to illustrate each subject. Fred sees himself as your guide on safari as you make your way through his book.

The book is divided into two parts — Getting Started, and Practical Programs. You are taken from the set-up procedure (including a description of each part of your system) through BASIC programming. Each chapter takes you further along in your becoming familiar with the T/S2068, and what you can do with it. One of the major differences between the Spectrum and the T/S 2068, the four display modes, are explained early on, but there is no explanation of the application of any but the normal mode, but as this is a basic guide, that is all you would expect. They keyboard is introduced in a brief program which gives you the opportunity to access most of the commands you will come across on the T/S keyboard.

The next section, entitled "The Basics of BASIC", is a tutorial on the use of Sinclair BASIC. Along the way, Fred has included a number of short programs which illustrate the procedures discussed. (Examples are the ESP tester which is used to demonstrate the use of the random number generator, and the digital timer which demonstrates the FOR. . . NEXT loop.) Each section has a few questions at the end to test how well you understand the material presented.

Part 2 gives you some practical programs and explains how they work. There are two sections in this part. The first one deals with home and personal use programming. Included in this section are "On Your Mark!", a footrace using graphics, colors and BEEPs; "Toll Totalizer", which is a digital timer to keep track of the cost of long distance telephone calls; and "Bio-Graph", which plots your biorythms. The second section, which is hobby programming, includes "Basic Banner" which allows you to print messages lengthwise along the T/S 2040 printer paper; "Collection Evaluator" which gives you a way to keep track of any collection you may have, "Sounds of Music?" which is a music maker program utilizing the Timex BEEP command; and "Long Distance Navigator", which may be used by pilots, boaters, or "ham" radio operators (such as Fred). All of these programs are fully documented, with each routine fully explained. Even if these programs are not useful to you as they are, the demonstration and description of the techniques involved are useful to anyone interested in programming his own applications. This book is a joy to read thanks to Fred's easy going style and humor, and ought to be in every T/S 2068 owner's library, from the novice to the experienced programmer.

The follow-up to this book, TIMEX SINCLAIR 2068 INTERMEDIATE/ADVANCED GUIDE, by Jeff Mazur, is a story in itself. When I first got my copy of the Guide, there were many things I had hoped to read about, but was referred to the upcoming Intermediate/Advanced guide for much of it. I contacted the publisher only to be told that the book has been removed from the publication calendar. However, if there were enough interest, it might be put back on the schedule. A letter writing campaign was launched, the decision was reversed, and it was finally published.

The Intermediate/Advanced Guide is designed to take you beyond BASIC — to make your programs run faster and do things otherwise impossible with BASIC. It also shows you a little more about how the computer works. Like its predecessor, it is divided onto two parts, the first being 'How Computers Work', and is a general guide to computers, not the T/S 2068 specifically. The parts of a computer and how they interrelate are discussed, as well as bits, bytes, and Boolean logic. The binary and hexidecimal number systems are also covered.

The final chapter in Part 1 is Exploring the T/S 2068 BASIC. The operating system is explained, and you are shown how the T/S 2068 handles variable storage, program storage, and the operation of the BASIC interpreter. Section B of Part 1 takes you inside the T/S 2068. The Z80 Central Processing Unit is explored, and the memory map is discussed. You are introduced to such things as bank switching and Input/Output facilities. A chapter discusses connecting the T/S 2068 to the outside world via the joystick connector, the DOCK connector, and the peripheral expansion connector. The programmable sound generator is next on the menu, and the SOUND command gets a thorough workout here. The last chapter in Part 1 describes the Video Display, and the four display modes on the T/S 2068. I must admit disappointment here, I had hoped that after reading this chapter I would be able to use the 64 column display mode. Unfortunately, although each mode is described with some technical material, there is no help given towards the utilization of any but the basic 32 column display mode.

Part 2 of the book is concerned with machine language programming. Chapters 9 through 13 are a tutorial on Z80 machine coding in general, with chapter 14 relating all of that to the T/S 2068. Chapter 15 is a guide to the routines built into the T/S 2068 ROM which can be called from your programs. Combined with the Beginner/Intermediate guide, these books form an excellent reference work for all users of the T/S 2068. Both are at the reasonable price of $9.95. The set would also make an excellent gift for that T/S 2068 owner whose birthday, or other occasion, is right around the corner.

Basics and Beyond

Another book just published by an author familiar to most Timex/Sinclair owners is T/S 2068 BASICS AND BEYOND by Sharon Zardetto Aker. Many of us recognize her as one of the most prolific contributors to Sync, and other publications which formerly covered the Sinclair computers. BASICS AND BEYOND is just what you would expect from Ms. Aker — a well written, easy to understand tutorial.

The book contains over 120 short progams written by Ms. Aker to illustrate the techniques she describes, and each program is accompanied by a line by line annotation of its structure. In fact, many of the commands just briefly mentioned in the Timex manual are illustrated in this book in such a manner that you can quickly understand how to use them in your own programs. In fact, not only are the most obvious uses pointed out, but many simple refinements are explored which can make your programs run more efficiently, and look more professional. (I have to admit that over the last few years, I have found many of Ms. Aker's tips extremely useful in creating many of my

own programs.)

Not only is pure programming discussed in this book, but you will be treated to enlightening discussions on music theory, logic, trigonometric functions, and so much more in a manner which makes you want to read the book from cover to cover. Twelve chapters and three appendices make up this book, and between its covers is a wealth of information not found in any other manual. Two full chapters are devoted to techniques including a section on program development and polishing. The chapter on graphics has the best description of user defined graphics I have seen anywhere. In fact, Appendix B gives you the necessary data required to produce 46 UDG's including card suits, balls, arrows, boxes, patterns, gobblers and building blocks. Although each chapter is a gem I was most impressed by the chapter on sound on the T/S 2068. The thirty pages of this chapter present the most thorough instruction of both the BEEP and SOUND commands. Although the BEEP command is relatively easy to master and use, the SOUND command is not. Ms. Aker, however, does not start with the presumption that you are familiar with music theory, or any of the other terms (such as envelope, register, etc.) Which are needed to utilize this capability. In fact, she does not assume any prior knowledge of music, but explains and illustrates everything you will need to know. When you finish this chapter, you should be able to create music (with three part harmony if you so desire) or sound effects to enhance your programs. To help you even further, Appendix C contains some charts with values to use in development of your programmed sound.

If I had to recommend one book for the T/S 2068 programmer, it would be this one. At $9.95 it is published by Scott, Foresman and Company (1900 East Lake Avenue; Glenview, Illinois 60025).

Powerful Projects

Another title recently published by the same publisher is POWERFUL PROJECTS WITH YOUR TIMEX/SINCLAIR by Jim Stephens. Unlike the books we have looked at so far, POWERFUL PROJECTS includes the T/S 1000 and 1500 in its descriptions of projects. This book is aimed at the Timex/Sinclair, owner regardless of which unit he has.

As you may guess from the title, this book is for those interested more in hardware development than software development. It is for the beginner who wants to learn something about computer electronics. A discussion of the Z80 CPU starts you off, leading into the physical requirements of the Timex computer. There is a discussion of control software and control devices, and then you are ready to start your first project. I/O ports are discussed and you are shown how to construct the 7475 Output port and the 8212 Input port. In chapter 4 you will build home control devices such as electrical and lighting controls, a three channel home appliance controller, a mail indicator and an automatic tape recorder controller. A chapter entitled Microbotics covers the topic including robotic parts, the robotic platform, the robotic arm and hand and robotic feedback. A speech synthesizer is included in the



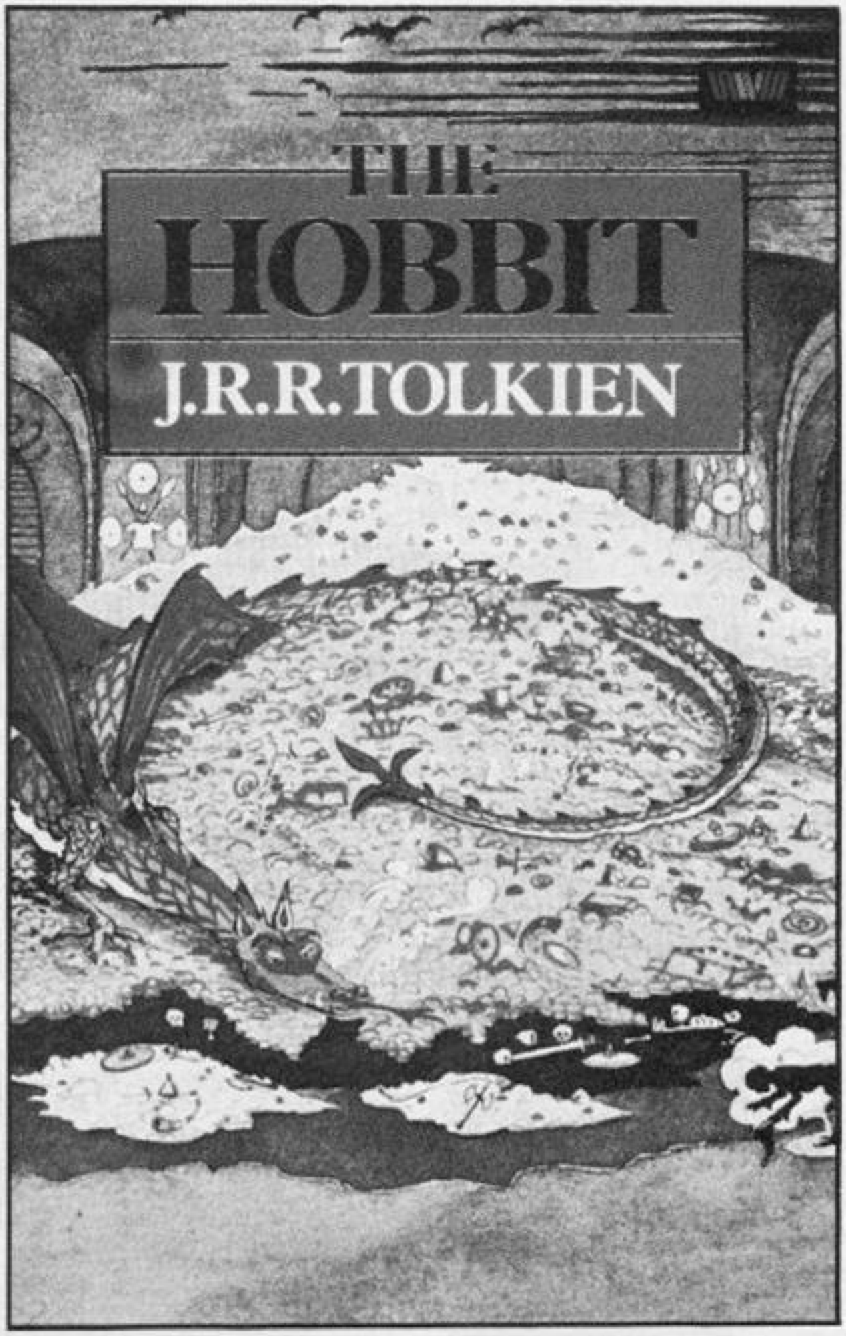
following chapter.

The final chapter has such advanced feedback projects as an analog to digital converter, a weather station, a moisture detector, wind indicator and temperature sensor.

Nine appendices (including a list of mail order suppliers) complete the book. Two of these are of interest particularly to the T/S 1000/1500 owner. They are Appendix H (A Device-Select Decoder for the TS1000 and 1500) and Appendix I (Converting 2068 Software for the TS 1000 and 1500).

This is not the complete list of books written specifically for the T/S 2068 computer, so look for more reviews in a future column.

A Visit To Melbourne House



Tucked away in a pleasant corner of Richmond, one of London's more tasteful suburbs, is Melbourne House, one of the longest established book and software publishers in the UK home computer market.

Following the release of the Spectrum in 1982, Melbourne House quickly built up a reputation for being THE top adventure company. However, the success of adventures such as *The Hobbit* and *Sherlock* has, to some extent, overshadowed the company's other achievements. So, now that they've gotten themselves a new MD, in the form of Geoff Heath (formerly of Activision), who plans to take the company into new areas, ZXC thought the time might be right to take a look at Melbourne House, past, present, and just a little way into the future.

The company's offices are in a suitably hi-tech building, with lots of dark, glass windows. Inside though, the offices are quite bright and spacious, and the open plan layout makes for a relaxed atmosphere. Tucked away in a corner known as Studio B, are the three in-house programmers. After a quick cup of coffee, I started to chat with Paula Byrne, the press and promotions officer.

Melbourne House was founded by Alfred Milgrom and Naomi Besen, and based, not surprisingly, in Melbourne, Australia. But, despite the fact that the Commodore 64 is the most popular home micro down under, when Melbourne House moved into the software field they began with the Spectrum.

The Hobbit

I doubt if there are many home computer owners that haven't heard of *The Hobbit*. Released in 1982 (originally for the Spectrum only), this was quite possibly the single most influential piece of software ever released in the UK market. Despite the hefty £14.95 price tag, which would be expensive even today, *Hobbit* sold and sold and sold, and continues to do so in its conversions for various machines. Its success was due to its innovative qualities — the complexity of the language handling, and the first ever screen graphics on an adventure game — and it not only put adventures onto the software map, but also set the standard for virtually everything that followed it. I noticed recently that Activision (Geoff Heath's former company, ironically enough) were taking out adverts implying that *Ghostbusters* had become the most successful title in the UK with sales of 300,000. It occurred to me that *Hobbit* must have sold just as well, and said so to Paula.

'We don't like to quote sales figures,' said Paula, almost grinning. 'But we're very happy with it.'

The Hobbit wasn't the only influential product to come out of Melbourne House in its early days, though. As Paula pointed out, 'we're very strong on utilities'. *Melbourne Draw* was the first graphics utility for the Spectrum, and was very sophisticated for its time. Only recently, with the benefit of nearly three years programming experience have other software houses produced utilities that can claim to be more advanced.

Then, there was *Abersoft FORTH*, one of the earliest implementations of that language for the Spectrum, and one which still sells well.

HURG was an early games designer, and a competition to design a marketable game with it brought forth a winner, 'but it's more suitable for the Commodore 64,' Paula told me, 'as it involves music. It's planned for the future, though there's no release date as yet.'

Despite the success of these utilities Melbourne House is unlikely to produce many more, as it seems that they tend to benefit other software houses as much as themselves. As Paula pointed out, *Melbourne Draw* was adopted by many software houses for designing games graphics. And, over the next year or so, games such as Mugsy, Sherlock and Hampstead, amongst others, all found ready acceptance in many people's collections (Hampstead was even voted game of the year by the BBC's own eminently trendy magazine, The Listener).

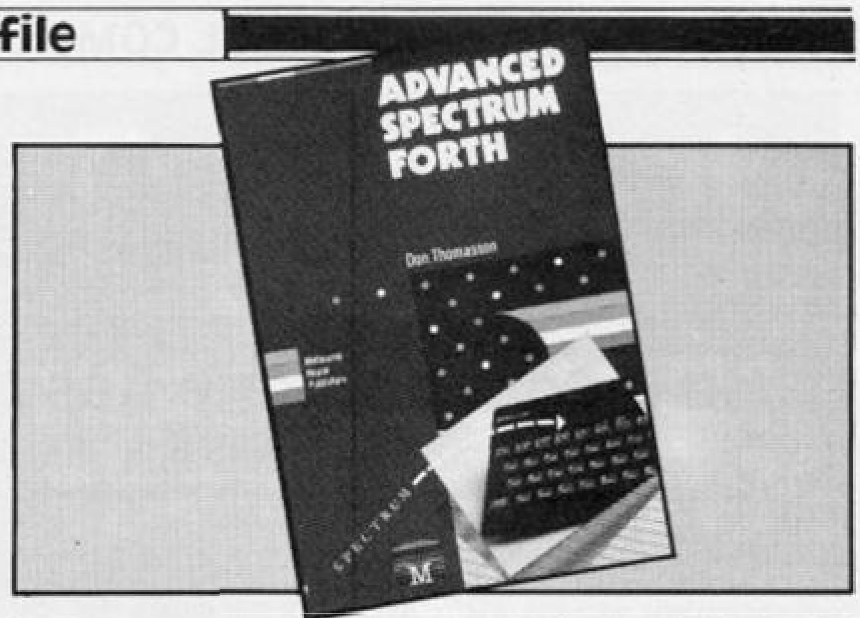
In addition to their software releases Melbourne House have also published a number of books dealing with various machines. Dr. Ian Logan's book on the Spectrum ROM disassembly is frequently referred to as the definitive guide to the inner workings of the machine. However, with their two latest books (see our reviews and competition elsewhere in this issue) they've now published eleven titles for the Spectrum, and Paula told me that they simply can't think of anything they haven't covered for the machine, so there may not be anymore Speccy books coming up for a while.

I asked how the company went about planning both books and software. Were titles commissioned, produced by an in-house team, or simply submitted by freelancers? It turns out to be a bit of each, the basic principle of in these matters being 'if you see a good idea, snap it up . . . we'll do anything if it's a good game.'

Ian Logan for instance, has been involved with the company right from the start, whereas others, such as the authors of *Hampstead*, sent in their program unsolicited. After a while, some freelancers become old hands and may be commissioned to do work such as conversions of games for various machines. The idea for *The Hobbit* though, came from within the company, from Alfred Milgrom himself, and Paula told me that Melbourne House has strong programming teams both in England and Australia.

That seemed a good point to





wander over to studio B, and meet the resident programmers, Nick Gilling and Alan Giles (who has about half a dozen pages in this issue of ZX all to himself). The third member of the team, Steven Cargill, was away that day.

Studio B

Just to stir up trouble, I asked the team what they thought of some of the machines currently on, or soon to be on, the market. 'The Commodore 64 isn't a particularly nice machine to work with,' Alan Giles said. 'But you can get good games on it. Somewhere in there are some good chips.'

The Amstrad also got a favourable response, and there may soon be some books and a *Hobbit* conversion for the machine. Mention of the QL didn't, I'm afraid, evoke much response, and there is no QL software planned for the immediate future. A possible QL *Hobbit* has been postponed indefinitely. It looks though, as if *Hobbit* may go to America in an IBM PC version.

The Spectrum is still a major machine as far as Melbourne House is concerned, and Paula reminded me that *Starion*, for the Spectrum only, reached No.2 in the software charts, and was only outsold by the enormously successful *Soft Aid* tape (which is available for both Spectrum and CBM 64). So they will be supporting the Spectrum for a long time to come. As for the 128K Spectrum, which Sinclair are still denying the existence of as I write this piece, Paula told me that they had no plans for it just yet, and would wait to see the machine and how well it sells.

But, back to the programmers. As well as the Studio B team, Melbourne House also works with Beam Software in Melbourne, who have teams for both graphics and music programming. How then, does it feel to be part of a team that is half here, and half 'down under'?

'I always feel I'm contributing,' said Alan, and the distance factor doesn't make them feel left out at all. The English team are linked to Australia via Telecom and Minerya (the Australian equivalent), and programs and code can be downloaded quite easily. Alan himself uses a TRS-80, and then downloads material into a Spectrum from there.

The time was coming for me to return to the ZX offices, but before I left I asked about the future of Melbourne House. Paula feels that the company has developed along with the industry itself, and that the 'big business' aspect was bound to take over.

'I started myself in a bedroom in Twickenham, but you've got to develop. You can't afford not to meet schedules and delivery dates.'

The company has its release schedule planned right up until Christmas, with titles such as *Terrormolinos, Way of The Exploding Fist* (or Exploding Cyst as one newspaper called it!) and, the one we've all been waiting for, the first part of *The Lord of The Rings*, for this autumn. The second and third parts won't be ready until next year, but they are sure to be eagerly awaited by adventurers everywhere. However, games like *Exploding Fist, Starion, Hellfire* and (for the '64) *Zim Salabim* have marked a move into the area of arcade games, something that Geoff Heath plans to continue in the future.

Paula's final comment, and a suitable point to end upon was:

'The main thing at Melbourne House at the moment is that we're very well known and strong on adventure, but now we want to become equally strong on arcade games.'



Q-Com Modem

It was touch and go for a while, but the Q-Com package has appeared at last.

When OEL went bust a while ago, QL owners looking forward to the company's promised communications package were left wondering when, if ever, they would be able to link their new computers into the outside world. Needless to say, Sinclair was equally upset as QL communications was seen as vital for the computer's marketability in the bottom end of the business market rather than just the top end of the games arena.

Tandata, however, came to the rescue and took over the production and marketing of the OEL product and the company's QL communications package is likely to have appeared by the time you read this article. According to Tandata, orders are already being shipped to those unfortunates who sent in their cheques to OEL before it went over the brink of bankruptcy.

Another QL communications package, the QL Bright Star manufactured by Modem House, recently hit the market and both products sell at just over £200. Although nobody expects two companies fighting over a fairly small market of QL owners to be the best of friends, there have been some hostile vibrations emanating from Modem House which suggest a competitive spirit bordering on the embittered. Tandata have yet to join in the mud-slinging, but if they do then micro mags should get a lot of fun copy.

I had hoped to review both products here but despite repeated promises over the last month Modem House's PR firm has yet to supply even the elementary Press Release package much less a modem for test. As Tandata supplied both pretty quickly, I see no reason to keep the company hanging around waiting for its review whilst I cultivate the patience of a statue trying to get even a few scraps of paper out of the competition.

From what I do know of the Bright Star, it appears a good product. It is an intelligent modem which, in some ways, is a more versatile communicator than Tandata's although — at least for the moment — it lacks some of other's more useful features. Bright Star ought to be a serious candidate for QL owners anxious to get on-line but, since all I've had out of Modem House so far is an odd collection of empty promises, self-praise and thoroughly bitchy remarks about Tandata, I can only recommed that potential buyers ensure that the company's after-sales services are better than its public relations.

Tandata's product consists of three separate units two of which are optional. The essential piece is Q-Connect, an intelligent module supporting QL communications from75 to 9600 baud supported by an extensive package of software on a QL microdrive. In my review package, the software looks to have been written in Basic — a slight inconvenience that I hope Tandata corrects in the final product — which obviously makes loading a longer and clumsier process than machine code. A fully machine coded version would give Q-Connect's software the same professional feel and appearance as the latest versions of the four Psion business packages that come free with each QL and give the owner five efficient, compatible and highly user-friendly programs.

Q-Connect is powered directly from the QL's external transformer unit and, in turn, powers the QL via a cable to the appropriate port on the back of the computer. This is a handy feature which avoids yet another wall plug but won't work if other peripherals are being powered from the QL. Other technical features include separate transmit and receive buffers, a 25-pin RS232 data port, and flow control. Q-Connect can be used with just about all asynchronous modems and communicates with the QL at 9600 baud via a cable connected to the computer's SER2 port at the back.

Extensive software features include a CET telesoftware downloader, a number of configuration menus including a mailbox, spooling for saving incoming data to microdrive, transmission of files on microdrive, a phonebook, off-line message preparation, error correction in user-to-user file transfer and VT100 terminal emulation for accessing databases such as Telecom Gold. There is an encryptation feature for encoding files before transmission. I haven't had much chance to play with the encoding procedure yet but it has a unique key per file system which makes it look pretty secure. The only drawback being that — if each file has its own key — you have to keep track of all the cyphers.

The encryptation feature is for transmission to other QLs and is essential if the QL is ever to be taken seriously as a business machine or for anybody who simply wants a bit of on-line privacy. To encode on Q-Connect, the computer will request a key and the user must then enter a string of up to thirty characters. The key must be entered again for decyphering which, since you have to keep all the different keys on record, makes things a bit clumsier than I would have liked.

Systems like the RSA use a public key for encoding, and a secret key for decoding, which are based on the immense difficulties involved in factoring very large numbers. There is no need to use different keys for different batches of data and the system is, for all practical purpose, unbreakable. British Telecom is developing cryptographic products based upon a similar procedure and I wish microcomputer companies would begin to look seriously at these sorts of user-friendly (more or less), but secure systems for encoding and decoding data.

The two other units which make up the Tandata QL communications package are Q-Mod and Q-CAll. Both are optional but are designed to work with Q-Connect's telesoftware. Q-Mod is the modem which communicates at 1200/75 baud (full duplex) for links to Prestel-format databases and 1200/1200 baud (half duplex) for QL to QL exchanges. The lack of a 300 baud facility is likely to be the chief criticism but, as Q-Connect can be used with virtually all asynchronous modems, the user will be able to extend the system's capabilities at some added cost. Q-Connect's software has a real-time clock facility for those who care to monitor the charges of accessing remote computers.

Q-Call is the auto-answer/auto-dial unit. To use it, a file name from Q-Connect's Phonebook is entered. The computer displays the number listed with that file name and Q-Call takes over for an automatic dial. Without Q-Call, the user is asked to dial the number displayed himself. An LED on Q-Call indicates that the auto-dial facility is functioning properly while an LED on Q-Mode shows that communication has been made and is in progress.

The three units are all smartly designed. Each is made of QL-style black plastic to match the computer in appearance. They are joined in a stacking system using vertical bus connectors. Q-Connect is the base with Q-Call fitted above it. Q-Mod rests on top of the stack and the whole package forms a compact, attractive unit without a lot of extra wires adding to the spaghetti at the back of the computer. My only real criticism is that the possibility of slightly careless handling runs a risk of damaging the bus connectors as they are the only links joining the three units. After all, even the best of us can make a mistake and a slightly securer joining ought to make the user rest easier.

The launch of the QL was marred by bad marketing, software bugs and the financial collapse of the company preparing the communications package. What promised to be a 'quantum leap' in home-micros floundered at the beginning, but by picking-up and carrying on with the OEL modem, Tandata have gone a long way towards getting the QL back on the path it should have been on from the start.

Brian Beckett

Q-Connect (plus software): £89.95
Q-Mod: £79.95
Q-Call: £49.95
Available from Tandata Marketing, Albert Rd North, Malvern, Worcs.
Tel: 06845 68421.

World Patents Pending

Slomo PACESETTERS

Joystick interfaces with a difference!

TWO NEW INTERFACES FOR THE SPECTRUM WITH NIDD VALLEY'S SLOW MOTION CONTROL BUILT IN



Both with an on/off button and LED to indicate slow motion, and fitted with through extender.

SPECTRUM PACESETTER

Kempston compatible with slow motion control **£14.95** incl. VAT

SPECIAL DEAL! Pacesetter with joystick *£21·90*

PROGRAMMABLE PACESETTER

Easy to program – enables ALL games to be played with joystick & speed control, by placing the control keys into the interface's 1K memory. Keeps a record of all your games so you only have to program once for each game. Compatible with microdrive and all popular joysticks.

Supplied with software and instructions **£29.95** incl. VAT

SPECIAL DEAL! Programmable with joystick *£34·90*

FLIGHTLINK JOYSTICK
Microswitched; very light positive action. Dual independent fire buttons. **£9.95**

QUICKSHOT II JOYSTICK
Popular joystick with two fire buttons and Autofire. **£10.95**

SPECTRUM KC A good quality reliable Kempston compatible interface.

WITH EXTENDER **£9.95** *SPECIAL DEAL!* with joystick *£17·90*

WITHOUT EXTENDER **£8.95** *SPECIAL DEAL!* with joystick *£16·90*

ASK FOR DETAILS OF OUR OTHER POPULAR PRODUCTS –
SLOMO – Hand held speed controller for BBC, CBM 64 or Spectrum **£14.95**
RESET EXTENDER – resets Spectrum without loss of power **£4.95**

SPECIAL DEALS – Prices apply either to Quickshot II or Flightlink joysticks – please specify which type required.
Prices include VAT and postage. To order, send cheque or P.O. payable to Nidd Valley Micro Products Ltd., to the address shown below (no stamp needed). Please allow 14 days for delivery – we do not bank your cheque until day of despatch.
MONEY BACK GUARANTEE IF NOT COMPLETELY SATISFIED.

NIDD VALLEY MICRO PRODUCTS LTD
Dept ZX 1085 FREEPOST, KNARESBOROUGH, N.YORKS. HG5 8YZ
Telephone: Harrogate (0423) 864488

SPECTRUM LIGHT PEN
ONLY £10.00



WORKS WITH ALL SPECTRUMS, CREATE PICTURES, SKETCHES, ANIMATIONS AND GRAPHS. PICK FROM MENU OF 16 ROUTINES. COMPLETE WITH INTERFACE CIRCUIT (PLUGS DIRECT IN TO USER PORT) AND SOFTWARE PROGRAM WITH COMPREHENSIVE INSTRUCTIONS, NO NEED TO KNOW ANYTHING ABOUT MACHINE CODE OR EVEN BASIC PROGRAMMING.

LIGHT PEN £10 inclusive VAT & PP

I enclose Cheque/Postal Order for £10 or Debit my Access/Barclay card/Visa

No.................................... Expiry Date ....

Name ....

Address ....

Signature ....

CUSTOM CABLES INTERNATIONAL LTD., Units 3 & 4 Shire Hill Ind. Estate, Saffron Walden, ESSEX. CB11 3AQ.

Do you want to learn machine code the easy way?

If you're serious about programming, you *need* machine code.

But, until now, it's been a hard slog through thick books and you've probably given up in despair before mastering the techniques.

Now there's a simpler answer: **THE PEKER MACHINE CODE COURSE.** Unlike other systems, the PEKER course is designed to take you simply, easily and quickly through ZX Machine Code in a way you will understand.

This is achieved by separate lessons which you learn at your own pace – each is designed with *you* in mind with examples at every stage.

Make no mistake. With the PEKER MACHINE CODE COURSE you'll be programming in fast machine code in minutes with a full understanding of how and why things work.

The course consists of 12 lessons at only £4.49 (plus P&P) per lesson. If you order the first six lessons you can choose a **FREE** gift from a selection of ZX software or a reduced price ROM-based Assembler.

If you really want to program like a professional, learn how to simply . . . with PEKER.

CHOOSE YOUR INTRODUCTORY GIFT FROM ANY OF THESE FANTASTIC ITEMS:
ORDER THE FIRST SIX ISSUES OF OUR MACHINE CODE COURSE AND YOU CAN CHOOSE ANY **TWO** OF THE ITEMS SHOWN AND THE CHANCE OF OUR REDUCED PRICE ASSEMBLER ROM.
ORDER ONE LESSON ONLY AND YOU CAN CHOOSE ANY **ONE** OF THE FIVE TAPES.
TAPE SOFTWARE:
ASSEMBLER · DISASSEMBLER · CHARACTER GENERATOR
TOOLKIT · FAST LOAD/SAVE.
OTHER ITEMS:
SCREEN PLANNER PAD · CHARACTER PLANNER PAD
PROGRAM PLANNER PAD.
SPECIAL OFFER
REDUCED PRICE ROM-BASED ASSEMBLER FOR ONLY £14.95
(NORMALLY £24.95)

POST TODAY – LOOK AT WHAT YOU CAN GET FOR YOUR MONEY:

Post to: PEKER COMPUTERS LTD., FREEPOST, Great Yarmouth, Norfolk NR30 3BR. (No stamp needed).

Name ____________________

Address ____________________

Please send me the following:

☐ **The first 6 lessons of your machine code course**
Total Cost @ £28.50 (inc P&P) I wish to claim the following items:

1. ____________ 2. ____________

3. ____________

If you have requested the ½ price ROM Assembler, please include an additional £14.95 for this item (P&P free)

☐ **The first lesson only of your machine code course**
Total cost @ £5.50 (inc P&P) I wish to claim my FREE software as follows:

1. ____________

TOTAL REMITTANCE ENCLOSED: £ ____________

Overseas orders should include an additional 20% on the prices quoted for *surface mail* post.

PEKER Computers

Light Screen Designer

Part Eight by Toni Baker



```
                        ORG E14E
2A615C    CHECK_INPUT   LD HL,(WORKSP)       Point CH_ADD to the start of
225D5C                  LD (CH_ADD),HL       the INPUT string.
CDFB24                  CALL 24FB,SCANNING   Evaluate or check the expression.
3A3B5C                  LD A,(FLAGS)
FD5637                  LD D,(FLAGX)         Check that the type of input
AA                      XOR D                (numeric or string) matches
E640                    AND 40               that expected.
2004                    JR NZ,INPUT_WRONG    Jump if not.
DF                      RST 18               A:= byte following expression.
FE0D                    CP "enter"           Return provided that the
C8                      RET Z                byte is "enter".
CF0B      INPUT_WRONG   RST 08/DEFB 0B       Use the error handling routine
                                             to generate a flashing "?".

                        ORG E168
E5        INPUT_ITEM    PUSH HL              Stack the registers.
C5                      PUSH BC
F5                      PUSH AF
AF                      XOR A                A:= 00.
CD0116                  CALL 1601,CHAN_OPEN  Select channel zero (lower screen).
ED5B615C                LD DE,(WORKSP)
2A635C                  LD HL,(STKBOT)       Clear the workspace without
CDE519                  CALL 19E5,RECLAIM_1  destroying the calculator stack.
21715C                  LD HL,FLAGX
7E                      LD A,(HL)            A:= (FLAGX).
E61F                    AND 1F               Clear bits relating to INPUT.
C1                      POP BC               B:= new values for FLAGX bits.
B0                      OR B
77                      LD (HL),A            Assign new bits of (FLAGX).
C1                      POP BC               C:= length of INPUT prompt.
C5                      PUSH BC
0600                    LD B,00
F7                      RST 30               Create sufficient room in workspace.
F1                      POP AF               A:= position of "C" cursor.
D1                      POP DE               DE: points to last byte of prompt.
EB                      EX DE,HL
EDB8                    LDDR                 Store prompt message in workspace.
13        I_PR_LOOP     INC DE
3D                      DEC A
```

Before I start on this article, I'd like to point out one error which occured in the last issue. It's in the CALC_CENTRE routine — in fact it's the very last instruction in this routine. The instruction currently reads *E9-JP (HL)*. This is incorrect. It should read *C9-RET*. The byte occurs at location *E12F* You can cure this error in BASIC by typing *POKE 57647,201*.

This article is all about ARCs. An arc is a curved line which is, in fact, a portion of a circle. In BASIC you can draw an arc using the DRAW statement. In our program we'll be able to do it in three different ways — take a look at Figure One. The first option, ARC RAD, is the same as in BASIC. It draws an arc from the origin cursor to the main cursor, and the angle through which the arc turns must be specified separately — in this case it must be INPUT. The remaining two options are different. ARC USING again draws an arc from the origin to the main cursor, but this time the centre of curvature must be specified with the marker cursor. Finally, ARC THRU draws an arc from the origin, through the marker, and onto the main cursor. Note that all arcs are drawn anticlockwise.

You should be aware of the shortcomings of the routines. Consider ARC USING for instance. For it to work properly, the centre of curvature should lie on an imaginary line dividing the other two points. If you specify a centre a long way from this imaginary line then you may not get what you expect. Remember also, that the arc is drawn anticlockwise — this means that if you get the main and origin cursors the wrong way round you'll also not get what you expect. For ARC THRU to work properly, the cursors should be the right way round — first the origin cursor, then the marker cursor, then the main cursor — in an anticlockwise curve. If you don't do this then the program will try to draw

```
20FC                    JR NZ,I_PR_LOOP
ED535B5C                LD (K_CUR),DE        Store position of "C" cursor.
FDCB377E                BIT 7,(FLAGX)
2026                    JR NZ,I_LINE_1       Jump if doing INPUT LINE.
2A5D5C                  LD HL,(CH_ADD)       Stack two of the system variables.
E5                      PUSH HL
2A3D5C                  LD HL,(ERR_SP)
E5                      PUSH HL
21A2E1    I_ERROR_RET   LD HL,I_ERROR_RET    Return here in case of
E5                      PUSH HL              syntax error.
ED733D5C                LD (ERR_SP),SP
2A615C                  LD HL,(WORKSP)       HL: points to INPUT string.
CDA711                  CALL 11A7,REMOVE_FP  Remove any five-byte-forms.
FD3600FF                LD (ERR_NR),FF       Cancel any sytax error.
CD2C0F                  CALL 0F2C,EDITOR     Input the expression.
FDCB01BE                RES 7,(FLAGS)        Signal "checking syntax".
CD4EE1                  CALL E14E,CHECK_INPUT Check for syntax errors.
1803                    JR I_CONT
CD2C0F    I_LINE_1      CALL 0F2C,EDITOR     Input the LINE string.
FD362200  I_CONT        LD (K_CUR hi),00     Cancel the "C" cursor.
CD1D11                  CALL 111D,ED_COPY    Print input string in lower screen.
ED4B825C                LD BC,(ECHO_E)       BC:= coords of last byte
                                             in lower part of screen.
CDD90D                  CALL 0DD9,CL_SET     Store this as print position.
21715C                  LD HL,FLAGX
CBAE                    RES 5,(HL)           Signal "edit mode".
CB7E                    BIT 7,(HL)
CBBE                    RES 7,(HL)           Cancel any INPUT LINE flag.
201C                    JR NZ,I_LINE_2       Jump if INPUT LINE.
E1                      POP HL               Cancel I_ERROR_RET address.
E1                      POP HL
223D5C                  LD (ERR_SP),HL       Restore error pointer.
E1                      POP HL
225F5C                  LD (X_PTR),HL        Store CH_ADD temporarily in X_PTR.
FDCB01FE                SET 7,(FLAGS)        Signal "evaluating".
CD4EE1                  CALL E14E,CHECK_INPUT Evaluate the expression.
2A5F5C                  LD HL,(X_PTR)
FD362600                LD (X_PTR hi),00     Cancel X_PTR.
225D5C                  LD (CH_ADD),HL       Restore CH_ADD to its former value.
180F                    JR I_EXIT
```

```
2A635C    I_LINE_2     LD HL,(STKBOT)
ED5B615C               LD DE,(WORKSP)
37                     SCF
ED52                   SBC HL,DE           HL:= length of INPUT LINE string.
44                     LD B,H
4D                     LD C,L
CDB22A                 CALL 2AB2,STK_STO_$  Stack onto calculator stack.
C36E0D    I_EXIT       JP 0D6E,CLS_LOWER   Empty the lower part of the
                                           screen and return.

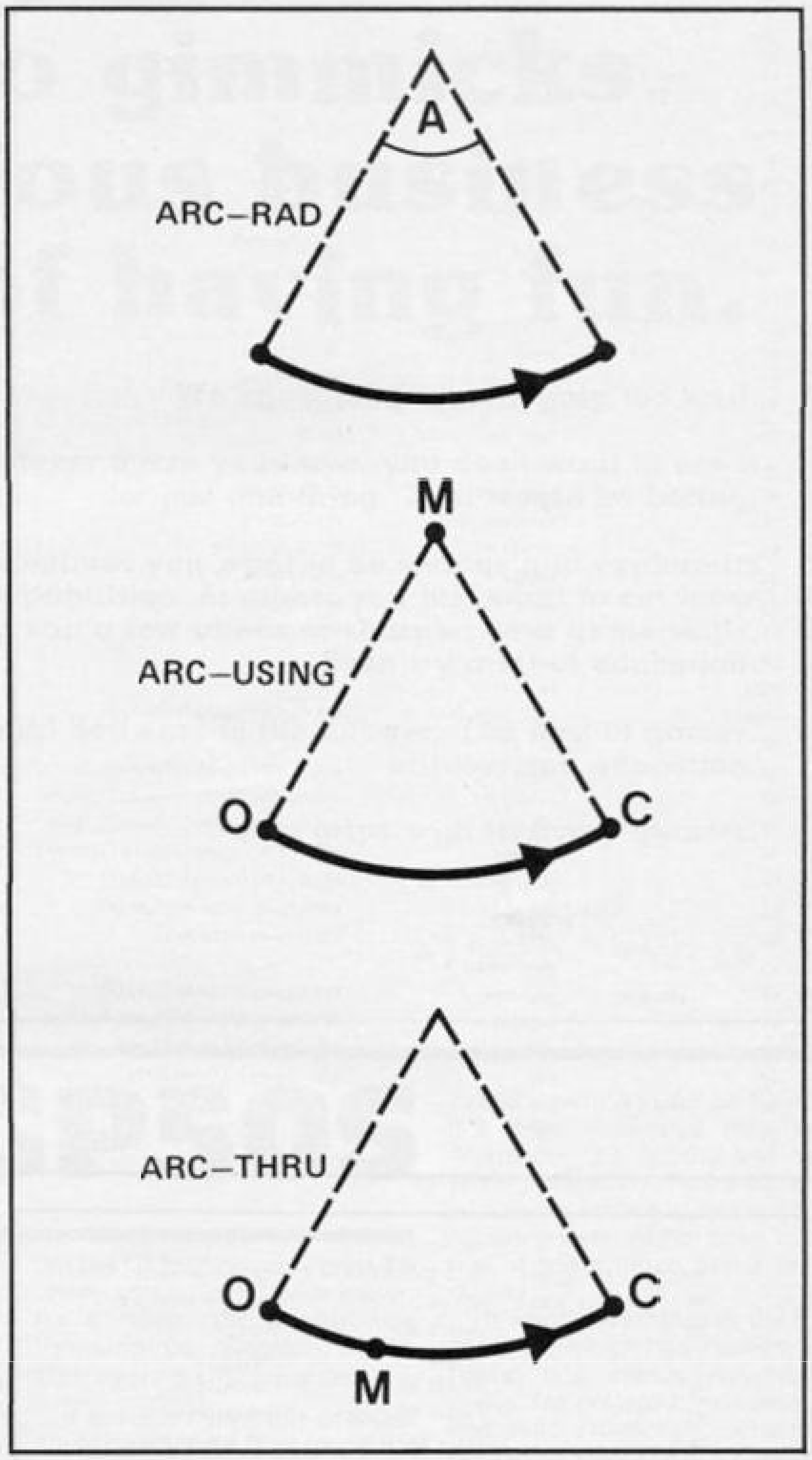
                       ORG E20A
ED4B0E5CDB START_LINE  LD BC,(ORIGIN+2)    BC:= coords of origin cursor.
CDEADE                 CALL DEEA,ADJUST_B  Adjust to ROM convention.
ED437D5C               LD (COORDS),BC      Store as "last point plotted".
C3A0ED                 JP EDA0,GET_CURSORS Get cursor coordinates into
                                           calculator memories and return.
```

```
                       ORG E218
CD0AE2    ARC_START    CALL E20A,START_LINE  Prepare to start arc.
EF                     RST 28                Activate the calculator.
E5                     recall M5             Cx
E1                     recall M1             Cx,Ox
03                     subtract              Cx-Ox
E4                     recall M4             Cx-Ox,Cy
E0                     recall M0             Cx-Ox,Cy,Oy
03                     subtract              Cx-Ox,Cy-Oy
38                     end calc              These are the first two
C9                     RET                   DRAW parameters.

                       ORG E224
28297A    ARC_HELP
A7CD                   DEFM ()*PI
```

```
                       ORG E229
C5        ARC_RAD      PUSH BC               Stack cursor coordinates.
CD18E2                 CALL E218,ARC_START   Initialise calculator.
CDCCDC                 CALL DCCC,MESSAGE     Print prompt message in
07                     DEFB 07               lower part of screen.
CDB6DD                 CALL DDB6,DR_CURSORS  Draw cursors on screen.
3E60                   LD A,60               Signal "input number".
010502                 LD BC,0205            B:= position of "C" cursor.
                                             C:= length of help prompt.
2128E2                 LD HL,ARC_HELP+4
CD68E1                 CALL E168,INPUT_ITEM  Input number of radians.
CDB6DD                 CALL DDB6,DR_CURSORS  Erase cursors from screen.
CD9423    ARC_END      CALL 2394,DRAW_ARC    Draw the arc.
C399ED                 JP ED99,CC_MOVE       Move the origin cursor and return.

                       ORG E248
C5        CALC_FN      PUSH BC
011E00                 LD BC,001E
F7                     RST 30                Make room in workspace.
11E3FF                 LD DE,FFE3
19                     ADD HL,DE             HL: points to start of space.
22685C                 LD (MEM),HL           Use space as calculator memory.
C1                     POP BC                B:= calculator function code.
EF                     RST 28                Use calculator to carry out
3B38                   DEFB 3B 38            required function.
ED5B615C               LD DE,(WORKSP)
2A635C                 LD HL,(STKBOT)
CDE519                 CALL 19E5,RECLAIM_1   Reclaim memory just used.
C3CB16                 JP 16CB,RESET_MEM     Restore original memory.
```



an anticlockwise curve from the origin cursor to the main cursor, along an imaginary circle which passes through all three points. Finally I should add that ARC THRU will not work if the three points are in a straight line (this is because it is then impossible to consider an imaginary circle passing through all three points).

The machine code in this article involves a lot of useful stuff — one particular example of which could be very, very, useful in other programs. It's a subroutine located at address E168, called INPUT_ITEM. It needs the subroutine CHECK_INPUT (here at address E14E) in order for it to work, but bearing that in mind it may stand alone. Its purpose is to simulate the BASIC INPUT statement. It can also supply a 'help' prompt. I can demonstrate this best by example. If you type INPUT A or INPUT LINE A$ you don't get any kind of help at all — just a 'C' cursor; if you type INPUT A$ however, you get two sets of quotes, with the 'C' cursor between the quotes. This 'quote quote' string is what I mean by a help prompt. A help prompt may be any sequence of characters which is supplied for you as part of the input string. The subroutine may be called providing that the registers *A, B, C, and HL* are properly initialised. *A* must contain *A0* (INPUT LINE string), 20 (INPUT string), or 60 (INPUT number); *C* must con- ▶

```
                  ORG E265
EF       ANGLE    RST 28               Engage the calculator.
A3                const pi/2           PI/2
E4                recall M4            PI/2,Cy
E2                recall M2            PI/2,Cy,My
03                subtract             PI/2,Cy-My
36                lt zero              PI/2,Cy<My?
30                not                  PI/2,Cy>=My?
0002              jump true ANGLE_1    PI/2
1B                negate               -PI/2
E5       ANGLE_1  recall M5            ±PI/2,Cx
E3                recall M3            ±PI/2,Cx,Mx
03                subtract             ±PI/2,Cx-Mx
30                eq zero              ±PI/2,Cx=Mx?
001D              jump true ANGLE_3    ±PI/2
E4                recall M4            ±PI/2,Cy
E2                recall M2            ±PI/2,Cy,My
03                subtract             ±PI/2,Cy-My
E5                recall M5            ±PI/2,Cy-My,Cx
E3                recall M3            ±PI/2,Cy-My,Cx,Mx
03                subtract             ±PI/2,Cy-My,Cx-Mx
05                divide               ±PI/2,(Cy-My)/(Cx-Mx)
38                end calc
0624              LD B,"atn"
CD48E2            CALL E248,CALC_FN    ±PI/2,ATN((Cy-My)/(Cx-Mx))
EF                RST 28
E5                recall M5            ±PI/2,ATN((Cy-My)/(Cx-Mx)),Cx
E3                recall M3            ±PI/2,ATN((Cy-My)/(Cx-Mx)),Cx,Mx
03                subtract             ±PI/2,ATN((Cy-My)/(Cx-Mx)),Cx-Mx
36                lt zero              ±PI/2,ATN((Cy-My)/(Cx-Mx)),Cx<Mx?
0005              jump true ANGLE_2    ±PI/2,ATN((Cy-My)/(Cx-Mx))
01                exchange             ATN((Cy-My)/(Cx-Mx)),±PI/2
02                delete               ATN((Cy-My)/(Cx-Mx))
3305              jump ANGLE_3
01       ANGLE_2  exchange             ATN((Cy-My)/(Cx-Mx)),±PI/2
31                duplicate            ATN((Cy-My)/(Cx-Mx)),±PI/2,±PI/2
0F                add                  ATN((Cy-My)/(Cx-Mx)),±PI
0F                add                  ATN((Cy-My)/(Cx-Mx))±PI
38       ANGLE_3  end calc
C9                RET
```

```
                          ORG E293
CD65E2   ANGLE_SWAP   CALL E265,ANGLE      Calculate angle.
EF       SWAP_MEMS    RST 28               Use the calculator.
E0                    recall M0            Oy
E1                    recall M1            Oy,Ox
E4                    recall M4            Oy,Ox,Cy
E5                    recall M5            Oy,Ox,Cy,Cx
C1                    store M1             (M1 now stores Cx)
02                    delete               Oy,Ox,Cy
C0                    store M0             (M0 now stores Cy)
02                    delete               Oy,Ox
C5                    store M5             (M5 now stores Ox)
02                    delete               Oy
C4                    store M4             (M4 now stores Oy)
02                    delete
38                    end calc
C9                    RET
```

```
                   ORG E2A5
CD22E0   ARC_USING CALL E022,TEST_MARKER  Return if marker not active.
C5                 PUSH BC                Stack coordinates.
CD18E2             CALL E218,ARC_START    Initialise calculator.
CD93E2   ARC_US_2  CALL E293,ANGLE_SWAP   Calculate main cursor angle.
CD93E2             CALL E293,ANGLE_SWAP   Calculate origin cursor angle.
EF                 RST 28                 Activate the calculator.
03                 subtract               X,Y,Ac-Ao
31                 duplicate              X,Y,Ac-Ao,Ac-Ao
37                 gt zero                X,Y,Ac-Ao,Ac>Ao?
0007               jump true ARC_US_3     X,Y,Ac-Ao
A3                 const pi/2             X,Y,Ac-Ao,PI/2
38                 end calc
34                 INC (HL)               X,Y,Ac-Ao,PI
34                 INC (HL)               X,Y,Ac-Ao,2*PI
EF                 RST 28
0F                 add                    X,Y,Ac-Ao+2*PI
38       ARC_US_3  end calc               Calculator stack now contains
                                          DRAW parameters.
CD38DF             CALL DF38,CANCEL_MARK  Cancel marker cursor.
C342E2             JP E242,ARC_END

                   ORG E2C5
CD22E0   ARC_THRU  CALL E022,TEST_MARKER  Return if marker not in use.
C5                 PUSH BC                Stack cursor coordinates.
CD18E2             CALL E218,ARC_START    Initialise calculator.
CDF8E0             CALL E0F8,CALC_CENTRE  Calculate centre of curvature.
CDA0E0             CALL E0A0,GET_CURCORD  Restore calculator memories.
111400             LD DE,0014
19                 ADD HL,DE
22655C             LD (STKEND),HL         Restore calculator stack.
EF                 RST 28                 X,Y,centre_x,centre_y
C2                 store M2               (M2 now stores centre_y)
02                 delete                 X,Y,centre_x
C3                 store M3               (M3 now stores centre_x)
02                 delete                 X,Y
38                 end calc
18CB               JR ARC_US_2
```

Figure 2. Modifications to the command addresses table.

DB4C	29	E2	DEFW E229,DRAW_ARC_RAD
DB5C	A5	E2	DEFW E2A5,DRAW_ARC_USING
DB6C	C5	E2	DEFW E2C5,DRAW_ARC_THRU



tain the number of characters in any help prompt (or one, if there is no prompt); *B* must contain the position within the prompt of the 'C' cursor, plus one (or one, if there is no prompt); *HL* is a pointer which must contain the address of an 'enter' character, which must be the last byte of a prompt string, if one exists. For an example of this, watch the way in which the subroutine is called in this article.

There is also a useful subroutine at address *E248* called CALC_FN, which executes a single calculator instruction (held in the *B* register) *without corrupting the calculator's memories.*

To make the routines given in this article work it is necessary to incorporate their addresses into the command addresses table. The necessary modifications are shown in figure 2.

A few words of explanation on some of the other subroutines given here. ANGLE (at address *E265*) calculates an angle. It effectively draws an imaginary line from the centre of curvature to either the origin or the main cursor, and calculates the angle between this line and the horizontal. SWAP_MEMS (*E296*) exchanges the origin cursor with the main cursor within the calculator memories.

The Light Screen Designer series is now almost complete — the eleventh part will be the final part. In the next issue I shall concentrate on the TEXT option — writing messages onto the screen. I leave you now with this month's programs. ■

Seikosha GP50S

This is an interesting option if you're thinking about getting yourself a printer, but are a bit wary of splashing out hundreds of pounds on a full blown, letter quality printer.

The GP50S is styled along the same lines as most of the Seikosha range, but scaled down a little. It is a dot matrix compatible with the ZX81, Spectrum and the USA Times/Sinclair machines. In other words, it plugs straight into the peripherals port at the rear of the computer, with no need for an interface of any sort, just like the old ZX Printer.

Setting it up was nice and easy, the manual is well produced and the illustrated instructions were very simple to follow. The first problem that I encountered came as I plugged the printer into my Spectrum's edge connector. The lead connecting the printer to the computer is hardly six inches long, and though the manual has a nice picture of the printer sitting side by side with the old rubbery Spectrum, my own machine is housed in a LO-Profile keyboard which is all of four inches wider than the original keyboard. As a result, it was physically impossible to have the two units side by side, there was no room to have them sitting back to back on my workspace, so the printer had to be moved around so that it was facing a fairly awkward angle. I think that an extender cable is probably *de rigeur* if you're going to buy this printer.

Once hooked up, the GP50S seemed to work perfectly well. There's no additional software required, all the Spectrum's built-in printer commands, LLIST, LPRINT, and COPY worked perfectly, and it handled graphics characters without any problems.

When I tried some wordprocessing using Tasword 2, I encountered a problem with carriage returns, due to the fact that Tasword uses 64 characters per line, which is fine if you're using a full size printer, but the GP50S can only print 46 characters on its 5" wide paper. As a result, the printer was throwing up question marks after every 64th character.

I assumed that this problem could be solved if I just found the necessary control code, and rang DDL's technical guys for help. The initial response was 'you're not trying to use it for serious wordprocessing are you?' No, of course I wasn't — the simple narrowness of the paper makes it unsuitable for that — but I did feel that it was worth trying it out just for experiments sake.

Now, although the GP50S may not be up to any serious word processing, I honestly don't think you can hold that against it. Certainly none of its competitors in the under £100 range can claim to be suitable for that task either. And if you're going to compare the GP50's performance with that of any of the other printers in that price range, then it will probably win hands down. At £69 plus VAT, it's more expensive than the Alphacom 32 (£55), and the same price as the Floyd 40, but is more versatile and, being a dot matrix printer, rather than thermal, it has a much better print quality.

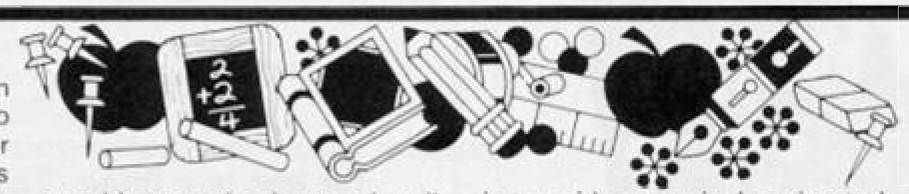
It's not as versatile as the Epson P40, which has a variety of fonts, and allows command codes for enlarged, condensed and bold characters, whereas the GP50S is by DDL's own admission, fairly unsophisticated and only allows about half a dozen control codes to be entered. What these do, I'm not quite sure about, as there's no mention of them anywhere in the manual. However, the GP50S is £30 cheaper than the P40.

So, if you want a fairly simple but reliable printer that isn't going to set you back £100+, and just want it for listings, screen dumps, and informal letters and notes (rather than high quality letters), then I think that the GP50S is probably just what you want (but don't forget the extension cable).

The Seikosha range is handled in this country by Data Distributors Limited, 5 Kings Rise Park, Ascot, Berkshire. ■

Spectrum lessons

Mike Edmunds looks at some more educational software.



Software houses specialising in educational software seem to have been off on their summer holidays for a while so there has been very little in terms of new releases for this issue. However, talking of sun, several rays of software hope *have* appeared and are on review this issue.

Happy Holidays

At last programmers have woken up to the fact that educational offerings deserve the same high quality as software for the games enthusiast. Quality, high-resolution graphics, good use of colour and sound appear to be essential for arcade-type games but, until recently, educational programs have, on the whole, fallen far short of those standards. Teachers and parents are becoming increasingly wary of 'educational' tags and today's software needs to be educationally sound, capable of wide usage across the curriculum, capable of supporting associated activities and, most of all, be good value for money.

This month's offerings go at least part way to satisfying these criteria, so, without further ado get out your buckets and spades and let's go for a 'Day at the Seaside'.

This seasonal contribution is part of Softlee's 'See-Hear' system. The idea is not entirely new to the Spectrum but is an exciting development for educational users. Softlee's system uses a stereo tape — one track providing the program, the other giving the audio soundtrack.

Loading this program was not easy, requiring a stereo output and careful volume setting, but, using the special lead supplied and having set up the headphones we can begin. The child (or children if you have multiple 'phones facility) listens to the soundtrack and responds to cues supplied both on the tape itself and within the program. The age level is 5+ and the soundtrack story is delightful for children of that age. The graphics are also impressive, illustrating the story and rewarding the correct spelling that the program requires.

Flashcards and word writing aids are included with the program, but although the whole package is well put together and simple to use it has a rather limited value used on its own. Described as an aid 'to beginning spelling' it can only run through the 21 given words in the order that they appear on the soundtrack. No variation is possible. It is useful for reinforcing the given words but it would have far more value used as a stimulus for the younger child to undertake some kind of topic work. Oral and language work are obvious follow-ups.

A great idea, superbly implemented but of limited educational value as a 'stand alone' program. (Also rather pricey!) Worth considering as a topic aid for younger children.

King Tut

Having spent some time digging in the sand, how about a search for King Tut's treasure while you're at it? Under the Mirrorsoft banner comes a double package of 'Educational Arcade Action for 5 — 11 year olds. 'King Tut's Treasure' and 'The Count' being the titles written by Soft Option.

These programs are designed to give Maths practice and skill reinforcement over a wide level of abilities, using arcade-format. As well as developing maths skills these programs call for a logical approach and simple mapping ability.

'King Tut' provides five options, ranging from shape matching to the matching of fractions with their decimal equivalent. A choice of objects, speed and number of hazards is also available to the teacher. Movement is via keyboard or joystick and the aim of the game is to guide your nicely animated archaeologist, Professor Diggins, around an ancient ruin. Equipped only with a metal detector (which emits a tone to indicate a 'find') you must discover the hidden treasure. As well as the necessary mapping there are various nasties zipping around which you will have to avoid if you are ever to gain the final reward!

In 'The Count' a search of Dracula's Castle is necessary before you meet and destroy the Count himself. All is not that easy however, as various maths problems have to be solved along the way. Counting, and the four Rules are covered with Easy and Hard options setting the number of problems.

Both games are superb in terms of graphics and colour and the problems seem ideally suited to the target age range. Given the fact that the pair of programs are available together for £7.95 they represent good value for money and a long awaited move toward higher educational programs. (A cause which, so far, has only been taken up by relatively few software houses!)

Look To The Sky!

Finally, after a day of sun, sand and excitement how about an evening stargazing?

From Skysoft comes a complication of programs on astronomy. The programs, tried and tested by teachers, are available from a main menu, with additional starmaps and constellations which can be loaded in. The options cover a perpetual calendar, a star tracker, options to view the moon's phases, paths of planets and satellites, the track of Halley's Comet, together with viewing the moons of Mars, satellites of Jupiter, rings of Saturn, planetary ephemeris, etc., etc.

The whole compilation is extremely thorough and although the documentation is rather limited there are on-screen explanations if required. The options are much too detailed to do justice to here, suffice it to say that this is one of the best programs of its type that I have seen. With monitor output it's almost like sitting in The London Planetarium!

The quality of colour, graphics and overall presentation is excellent (the text layout has been designed using 'Letset' by Eclipse) and the capabilities of the Spectrum have been fully utilised. The Starmap option in particular is first class, including seasonal changes. An additional plus is the ability to print out the display.

No indication of age-range is given, but for the more able junior child through to the experienced astronomer I cannot see a better value program becoming available for the Spectrum (even though it has one spelling mistake!). At £3.95 it represents outstanding value. Wholeheartedly recommended by someone who, as yet, cannot tell Regulus from Capella!

In conclusion I think that this month's programs typify the sort of material that many teachers with Spectrums have been searching for. If Summer or even Autumn days should happen to be wet, then power up your computer and settle down with some these programs!

A Day At The Seaside (£7.95, leads £1.95), Softlee Systems, 35 Windy Arbour, Kenilworth, Warwickshire.

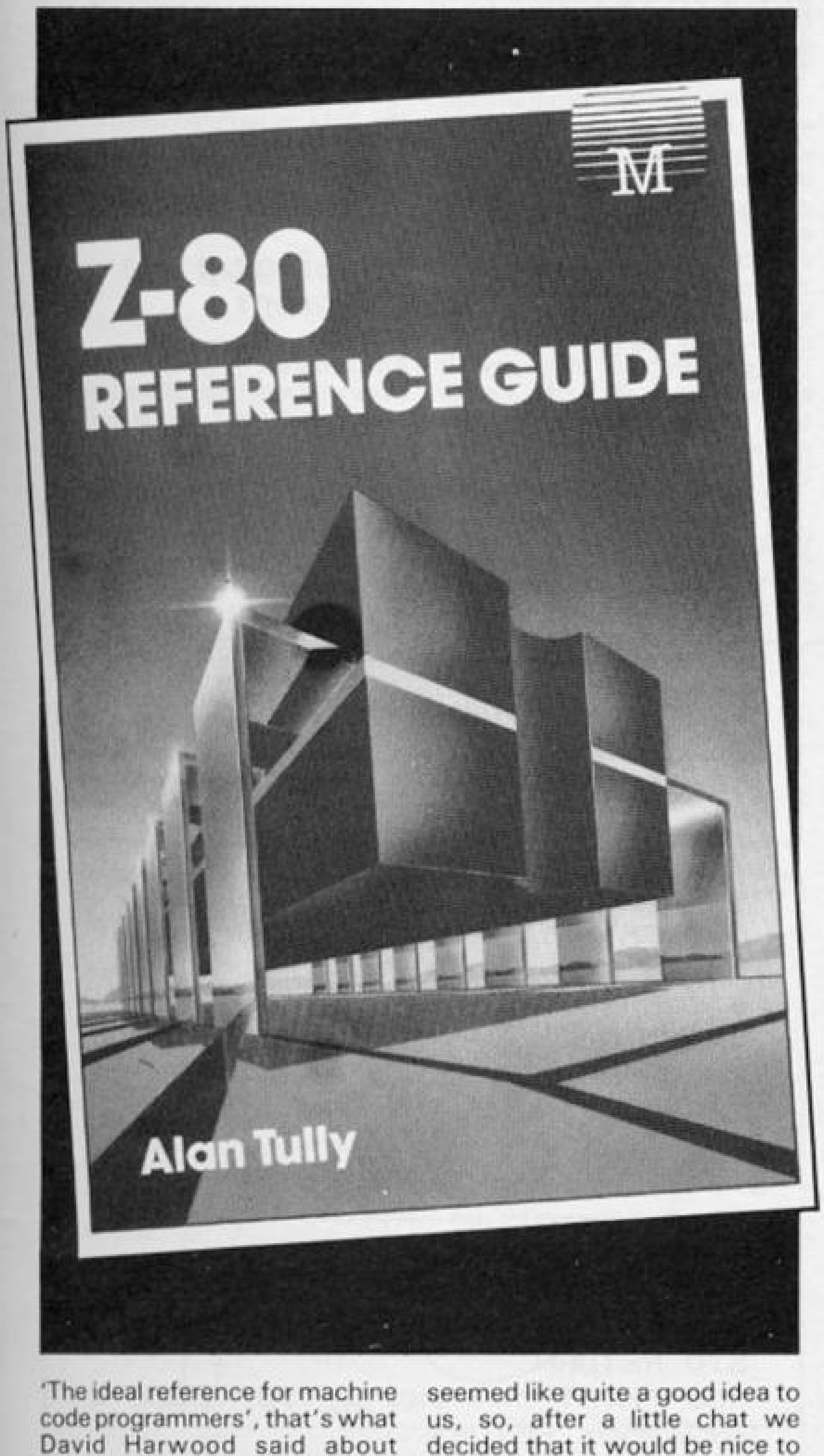
Ancient Quests (£7.97), Mirrorsoft, Mirror Group Newspapers, Holborn Circus, London.

Astronomy Compilation (£3.95), Skysoft, 38 Norton Ave, Surbiton, Surrey.

All programs are for the 48K Spectrum/Spectrum Plus.

Z-80 Reference Guide

We've got ten copies to give to aspiring programmers.



'The ideal reference for machine code programmers', that's what David Harwood said about Melbourne House's latest book, *The Z-80 Reference Guide*. And while I was visiting Melbourne House just recently, Paula Byrne said 'Why don't we arrange a competition sometime?' That seemed like quite a good idea to us, so, after a little chat we decided that it would be nice to give away some copies of the *Guide*. All you have to do to enter the competition is to answer a few questions about Melbourne House, fill in the entry form and stick it into the post to us. And, to make thing even easier for you, all the answers to the questions are contained in our profile of Melbourne House which is somewhere else in this copy of *ZXC*.

The rules

- This competition is open to all UK and Northern Ireland readers of ZX Computing, except employees of Argus Specialist Publications, their printers and distributors, employees of Melbourne House Publishers, and anyone associated with the competition.
- All entries must be postmarked before the 30th November and sent to the Editorial address.
- No correspondence will be entered into with regard to the results, and it is a condition of entry that the Editor's decision is final.
- The winners will be notified by post, and the results published in a future issue of *ZX Computing*.
- Entries must be on the coupon provided, but as long as each entry is on the correct coupon, there is no limit to the number of entries that can be made by an individual.

The questions

1) Name the two founders of Melbourne House.
2) Name two of Melbourne House's adventure games.
3) Name two of their arcade games.
4) What is the name of Melbourne House's software team in Australia?
5) What is the name of the forthcoming adventure game, based upon a novel by J.R.R. Tolkein?
6) As a tiebreaker, tell us (in 25 words or less) why you would like a copy of the Z80 Reference Guide.

There, nothing to it.

Binatone Competition

There was a good response to this competition, but in the end the six winners were:

Mr. I. Taylor, Devon; Andrew Magnay, Merseyside; Nadeem Walayat, Sheffield; David Wright, Lancashire; R.&P.R. Schofield, Somerset; J.F. Butler, Surrey.

All the above should have received their prizes by the time you read this — happy loading!

If you want to delve into the depths of machine code, then this is the book for you! Just fill in the answers to the questions in the space provided below, and then send the coupon to the address at the bottom of the coupon.

Question 1) . . . .

Question 2) . . . .

Question 3) . . . .

Question 4) . . . .

Question 5) . . . .

Question 6) . . . .

Name . . . .

Address . . . .

. . . .

Send this coupon to: Z80 Competition, ZX Computing, 1 Golden Square, London W1R 3AB.

After the Games

Regular contributor and sage, John Wase, muses on some practical ways of answering the perennial question, "But what does it DO?"

All right, I know that there are some superb Spectrum games to play, and it's even more fun hacking into them to find the infinite lives POKE. But just POKEing games can be like tinkering with a car to see how it works; I prefer driving it to get somewhere. After all, the Spectrum is more than just a games machine, it is an extremely well designed and powerful 8-bit micro, with bags of memory, although, regrettably, the famous rubber buttons have resulted in a certain lack of credibility. Nonetheless, people out there are actually applying it to everyday jobs, so I set out to discover what some of my friends were using it for. The uses ranged from the prosaic to the exotic. Here's a couple......

Number crunching

Chemical engineers often have to sovle complicated algebraic equations which do not have analytical solutions. Our Bill McManamey wrote a simple program, (Fig 1), involving a search method of solution to do this.

Decreasing increments

For equations of the type $f(x) = 0$, the solution is obtained by putting $f(x)$ equal to d and changing x until d is acceptably near to zero. At first, relatively large increments of x are used, then, nearer the answer, the increment is reduced. In other words, you are asking the computer to guess the answer, and then see how good the guess is so that it can make a better one. There are sufficient REMarks to make most of the program self explanatory, but, as the diagrams show, (Fig. 2), adding increments of a can cause d to change sign as zero is crossed; either from positive to negative, or *vice versa*. Line 80 enables the program to deal with d either increasing or decreasing as x increases. As it is a comparatively straightforward program, it stops as soon as it has a solution. If more solutions are suspected, (Fig 3), then it has to be restarted.

If you are not sure about it, just think of the equation $2x^{0.8} - \ln(3x) - 1.7 = 0$. Type in the listing, RUN it and ENTER the sample values for x, a, b and the equation, provided in the REM statement in line 600. Sometimes it will take a bit of time, but in this case it will take only four seconds to come up with the solution $x = 2.0107$. Try doing that by hand!

Storing shapes

The next example, by Susan Laflin of our Computer Studies Department (of Birmingham University) is much more unusual; moreover the principles upon which it is based are of very wide application. Using her program, she can INPUT a two-dimensional shape, DRAW it on the screen, then smooth the data as required before SAVEing on microdrive. Data for many shapes can therefore be stored.

Shaping up the program

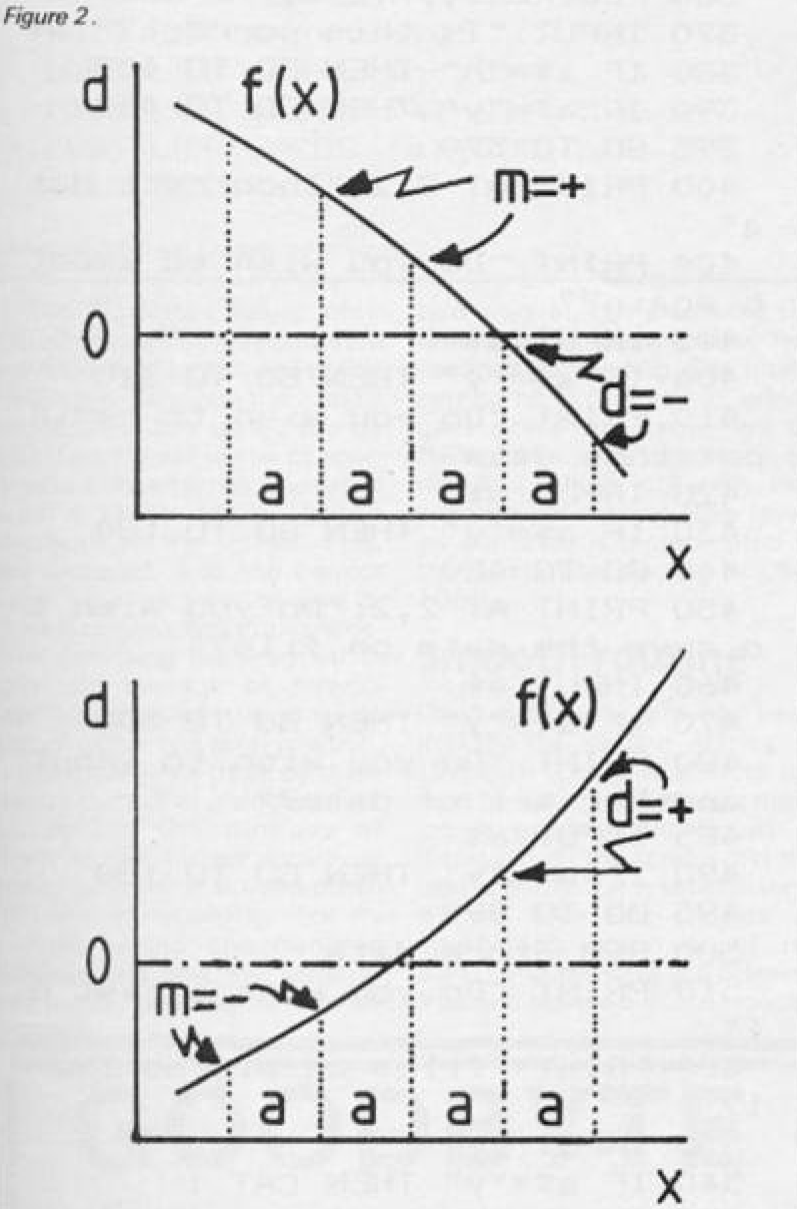
This program, (Fig. 4), is one of a suite of three (the others will be mentioned later), and is written in BASIC. This is slow to run, but then one operation of the tracer with which it is used is even slower, so that is of little consequence. Moreover, it allows the user to alter the SAVE routines without too much difficulty.

Figure 1.

```
  5 PRINT "**DECREASING  INCREM
ENT METHOD**"
 100 REM Main loop
 105 PRINT "x = initial trial va
lue,         (which must be less
than the     correct value),
             a = initial incremen
t of x,      b = acceptable value
 of d)."
 110 INPUT " x = ",x," a = ",a,"
 b = ",b
 115 GO SUB 400
 120 GO SUB 500
 125 REM In case the first guess
 is correct - unlikely!
 130 IF ABS d<b THEN GO SUB 550:
 STOP
 135 REM See 175
 140 LET m=d
 150 LET x=x+a
 160 GO SUB 500
 165 GO SUB 550
 170 IF ABS d<b THEN STOP
 175 REM This enables the
program to deal with d either
increasing or decreasing as x
increases
 180 IF SGN d=SGN m THEN GO TO 1
50
 190 LET x=x-a
 200 LET a=a/10
 210 GO TO 120
 410 REM Input equation
 420 PRINT "Rearrange your equat
ion to the  form f(x) = 0"
 430 PRINT "Press any key to con
tinue"
 440 PAUSE 0: CLS : PRINT "ENTER
 THE STATEMENT              500 L
ET d=f(x)                  (wher
e f(x) is your equation in x, eq
ual to zero).              THEN
ENTER CONTINUE": STOP
 500 LET d=function of x equal t
o zero
 510 RETURN
 550 PRINT "x=";x,"d=";d: POKE 2
3692,255: RETURN
 600 REM  x=1, a=0.2, b=0.00001,
 f(x) = (2*(x^0.8))-(LN (3*x))-1
.7
```

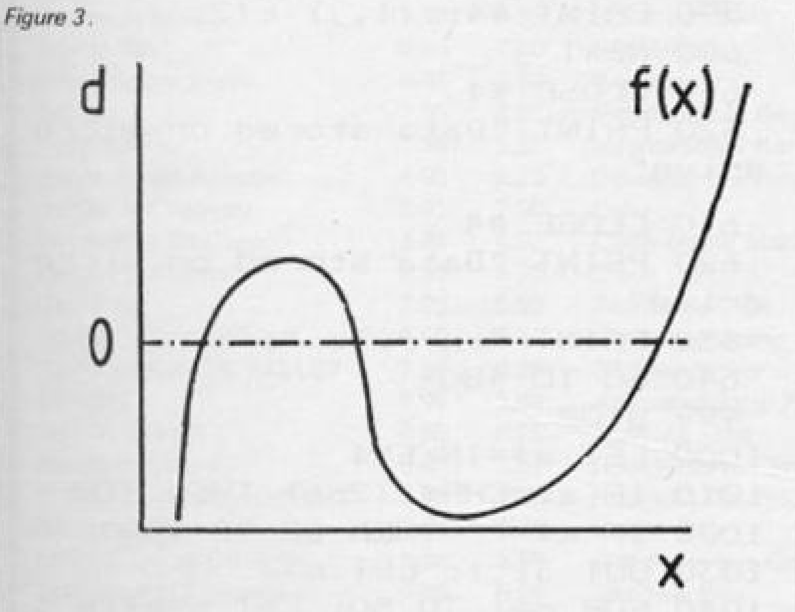
Listing of program to solve equations by the search method.

Figure 2.



Graphs showing how d changes sign as f(x) passes through zero: lines 140 and 180 enable the program to deal with d either increasing or decreasing.

Figure 3.



Graph showing how more than one solution is possible; (there are three here). The program was not intended to cope, but can be restarted with a new x greater than the value it has just found if this situation is suspected.

The hardware

Susan finds an ordinary rubber-keyed Spectrum quite satisfactory. Profiles are drawn on paper and the data digitised. Whilst, in theory, profiles could be drawn on graph paper, allowing one to read off the coordinates and type them in, in practice this is slow and tedious, and accuracy rapidly diminishes as one becomes tired. Of the three digitisers available, the RD-Tracer, the Grafpad and the Touchmaster, the latter two use a stylus to indicate the position and cost around £150 for A4 versions. The A4 RD-Tracer is only around £60, (£70 for an A3 version), so the A4 RD-Tracer was chosen. Whilst RD Laboratories are unfortunately no longer with us, the tracer is still sold by Kane-May and advertised by most journals.

It is convenient to attach the tracer to a small cork notice board by its clamp; for each INPUT session a profile is pinned to the board. The area used is about A4 in size: the paper is always attached to roughly the same place, but this is never, of course, exactly the same, so the first section of the program reads in the four corners of the profile area and calculates the scaling factors to map the rectangle onto the screen used by the Spectrum. This is done once for each session.

In goes the profile

This part of the program starts at LINE 180. The cross-wires are placed over each point to be INPUT and any letter on the keyboard is pressed. One hand is used to move the tracer whilst the other is on the keyboard. As the profile is INPUT, values are stored in the array *c* and a copy is displayed on the screen. INPUT is stopped automatically when 200 points have been entered, and can be stopped before this by pressing a number instead of a letter.

Figure 4.

```
  5 REM **********************
  6 REM ****PROGRAM TRACE1****
  7 REM ** @ S. LAFLIN 1985 **
  8 REM **********************
 10 DIM C(2,205)
 15 LET p=PI/400
 20 REM  DRAW SCREEN
 30 PAPER 4: BORDER 7: INK 0: C
LS
 40 PLOT 48,40: DRAW 0,86
 50 PLOT 48,120: DRAW 128,0: DR
AW 0,-80
 60 PLOT 184,40: DRAW -136,0
 70 PRINT AT 5,6;"y"
 80 PRINT AT 16,23;"x"
 90 PRINT AT 17,25;"O"
100 PRINT AT 18,6;"Position Tra
cer at point"
110 PRINT "corresponding to tha
t indicated"
120 PRINT "and press any key"
130 FLASH 1: PRINT AT 16,6;"A"
135 GO SUB 1000: FLASH 0: PRINT
 AT 16,6;"A"
136 LET xO=x: LET yO=y
140 FLASH 1: PRINT AT 16,21;"B"
145 GO SUB 1000: FLASH 0: PRINT
 AT 16,21;"B"
146 LET yO=0.5*(yO+y): LET xmax
=x
150 FLASH 1: PRINT AT 7,6;"C"
```

```
 155 GO SUB 1000: FLASH 0: PRINT
 AT 7,6;"C"
 156 LET ymax=y: LET x0=0.5*(x0+
x)
 160 FLASH 1: PRINT AT 7,21;"D"
 165 GO SUB 1000: FLASH 0: PRINT
 AT 7,21;"D"
 166 LET ymax=0.5*(ymax+y): LET
xmax=0.5*(xmax+x)
 170 LET xs=1
 171 IF xmax=x0 THEN GO TO 175
 172 LET xs=255/(xmax-x0)
 175 LET ys=1
 176 IF ymax=y0 THEN GO TO 180
 177 LET ys=175/(ymax-y0)
 180 PAPER 7: BORDER 4: CLS
 181 PRINT "Commence plotting."
 182 PRINT "Press any letter to"
 183 PRINT "input next point."
 184 PRINT "Plotting ends when"
 185 PRINT "205 points have been
"
 186 PRINT "input, or when a"
 187 PRINT "digit is pressed."
 190 GO SUB 1000: CLS : LET i=1
 200 LET c(1,i)=xs*(x-x0)
 210 LET c(2,1)=ys*(yjy0)
 220 IF c(1,i)<1 THEN LET c(1,i)
=1
 230 IF c(1,i)>255 THEN LET c(1,
i)=255
 240 IF c(2,i)<1 THEN LET c(2,i)
=1
 250 IF c(2,i)>175 THEN LET c(2,
i)=175
 260 PLOT c(1,i),c(2,i)
 262 IF i>2 THEN GO TO 270
 264 FOR i=2 TO 3: LET c(1,i)=c(
1,i-1)
 266 LET c(2,i)=c(2,i-1): NEXT i
 268 LET i=3
 270 LET i=i+1: IF i>203 THEN GO
 TO 300
 280 IF a$<="9" THEN GO TO 300
 290 GO SUB 1000: GO TO 200
 300 LET num=i+1
 304 FOR i=num-1 TO num
 306 LET c(1,i)=c(1,i-1): LET c(
2,i)=c(2,i-1): NEXT i
 310 PAPER 7: BORDER 3: CLS
 320 LET num=num-5: IF num<2 THE
N GO TO 400
 330 FOR i=1 TO num
 340 LET c(1,i)=0.2*(c(1,i)+c(1,
i+1)+c(1,i+2)+c(1,i+3)+c(1,i+4))
 350 LET c(2,i)=0.2*(c(2,i)+c(2,
i+1)+c(2,i+2)+c(2,i+3)+c(2,i+4))
 360 PLOT c(1,i),c(2,i): NEXT i
 370 INPUT "Is this correct?";a$
 380 IF a$="n" THEN GO TO 400
 390 IF a$="y" THEN GO TO 450
 395 GO TO 370
 400 PRINT AT 2,2;"Incorrect dat
a"
 404 PRINT "Do you wish to smoot
h again?"
 405 INPUT a$
 406 IF a$="y" THEN GO TO 310
 410 PRINT "Do you wish to re-in
put this data?"
 420 INPUT a$
 430 IF a$="y" THEN GO TO 180
 440 GO TO 480
 450 PRINT AT 2,2;"Do you wish t
o save the data on file?"
 460 INPUT a$
 470 IF a$="y" THEN GO TO 500
 480 PRINT "Do you wish to input
 another set of data?"
 485 INPUT a$
 490 IF a$="y" THEN GO TO 180
 495 GO TO 999
 500 REM Store data
 510 PRINT "Do you want a list o
f"
 520 PRINT "files on microdrive
1?"
 530 INPUT a$
 540 IF a$="y" THEN CAT 1
 550 INPUT "Name of new datafile
";f$
 560 OPEN# 4,"m";1;f$
 570 PRINT #4;num
 580 FOR j=1 TO num
 590 PRINT #4;c(1,j) c(2,j)
 600 NEXT j
 610 CLOSE #4
 620 PRINT "Data stored on micro
drive"
 610 CLOSE #4
 620 PRINT "Data stored on micro
drive"
 630 PRINT "in file ";f$
 640 GO TO 480
 999 STOP
1000 LET a$=INKEY$
1010 IF a$=CHR$ (266) THEN STOP
1020 IF a$="" THEN GO TO 1000
1030 OUT 31,1: LET m=0
1040 FOR o=1 TO 50: LET m=m+IN 3
1
1045 NEXT o: LET m=0.02*m
1050 OUT 31,0: LET n=0
1060 FOR o=1 TO 50: LET n=n+IN 31
```

```
1065 NEXT o: LET n=0.02*n
1070 LET a=p*(m-10)
1080 LET b=a-p*(n-10)
1090 LET x=SIN (a)-SIN (b)
1100 LET y=COS (a)-COS (b)
1110 RETURN
```

Listing of program for storing shapes.

The RD-Tracer inputs information via port 31; the software provided used an average of five readings to calculate the x and y coordinates. She tested this by INPUTing the same pair of coordinates a dozen times: the result was a large black patch displayed on the screen. This, she deduced, was the correct value, plus an error of two or three pixel positions in any direction. Changing the program to take the average of twenty readings reduced the error to plus or minus one pixel position, but this still left a three by three square instead of a single point. Increasing the number of readings still further improved the accuracy but slowed down the rate of recording. For instance, taking one hundred readings reduced the error to half a pixel position, (a two by two square), but increased the time to an intolerably slow five seconds per reading. She finally settled on fifty readings which gave similar accuracy (two by two pixel square maximum error), but which took only two seconds per reading. This revised software is incorporated in the subroutine starting at LINE 1000.

Smooth routine

Once sufficient data has been INPUT, the routine staring at LINE 300 is used to smooth the data by taking a five point moving average. The result is displayed on the screen and the user is asked if it is satisfactory; if not, then the choices of smoothing again or of re-INPUTting the data are offered. You will have no doubt noticed that although I mentioned earlier that 200 points were INPUT, the routine appears to use more. The first point is input three times and the last two to allow the smoothing routine to work properly. Duplicate points are subsequently discarded.

For SAVEing data, a choice of cassette, microdrive or wafadrive was available. Most profiles require between a hundred and two hundred coordinate pairs and for this purpose cassette was unacceptably slow. The Wafadrive has potentially a larger capacity than microdrive, but uses the same port on the edge connector as the RD-Tracer. The routine is therefore set up for microdrive. Each microdrive cartridge has room for about 40 complete profiles and about ten times as many fitted curves (see later).

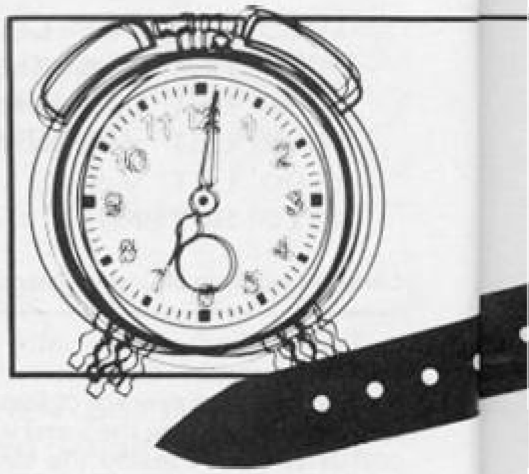
The SAVE routine is located at LINE 500 onwards, and could easily be changed to accommodate, for instance, a floppy disc system provided that it is compatible with the RD-Tracer.

Applications

Susan's program is used, surprisingly, in an archaeological application. Pottery profiles are read in, smoothed and stored for later comparisons. Later programs deal with the fitting of a B-Spline curve to the data; (thus many profiles can be stored as relatively small amounts of data), and then comparison of the profiles using the parameters of the fitted curves. Thus pieces of pottery which match are readily identified, as are similarities in overall shapes. However, I am sure that this is not the only application for this principle: what could you use it for?

[1] *This method is described in Jenson, V. G. & Jeffreys, G. V., "Mathematical Methods in Chemical Engineering", 2nd edition, Academic Press, London, 1977, p. 406. It is for finding* x *in equations of the type* $f(x) = 0$. *The solution is obtained by putting* $f(x) = d$ *and changing* x *until* d *is acceptably near to 0. At first, relatively large increments of x are used, then, nearer the answer, the increment is reduced. In some cases a program using linear interpolation in the interval a (or Newton's method) can be quicker; this is discussed in Jenson & Jeffreys.*

Clocking On



Carol Brooksbank keeps track of time with this useful machine code program.

When you are planning a game program with an 'against the clock' element, it would be useful to be able to display a clock running continuously on the screen. The Spectrum has its own built in clock. The problem is how to display it, and keep it running, on a machine which is not equipped for multi-tasking.

The Spectrum handbook has a BASIC program which, by PEEKing the three FRAMES system variables, and calculating the time elapsed from the values held there, produces an accurate clock. The trouble is that while the Spectrum is doing that it is not doing anything else. It seems that you can have the clock or your game, but not both. But it can be done, if you know a little about the interrupt modes.

Interrupts

In the normal state of Spectrum affairs, it stops what it is doing 50 times per second and performs the instructions in the ROM subroutine at 0038h — updating the FRAMES variables and scanning the keyboard. This is Interrupt Mode 1. (IM1) But there is another Interrupt Mode, IM2. (Actually there are two, but this is the one which concerns us here.)

In IM2, you can direct the Spectrum, at an interrupt, to your own subroutine in the RAM, and give it other tasks to perform. It is a powerful instruction, but must be handled with great care, as there are a number of conditions which must be met if you want to avoid everything going horribly wrong.

First, your interrupt subroutine must be short. There are 50 interrupts per second, so if your instructions take longer than 1/50sec. to perform, the program will crash. The Spectrum will not return from the interrupt subroutine before it is called again, and it will do nothing but repeat the first part of the subroutine over and over. Although the Spectrum can per-

CLOCK CODE LISTING

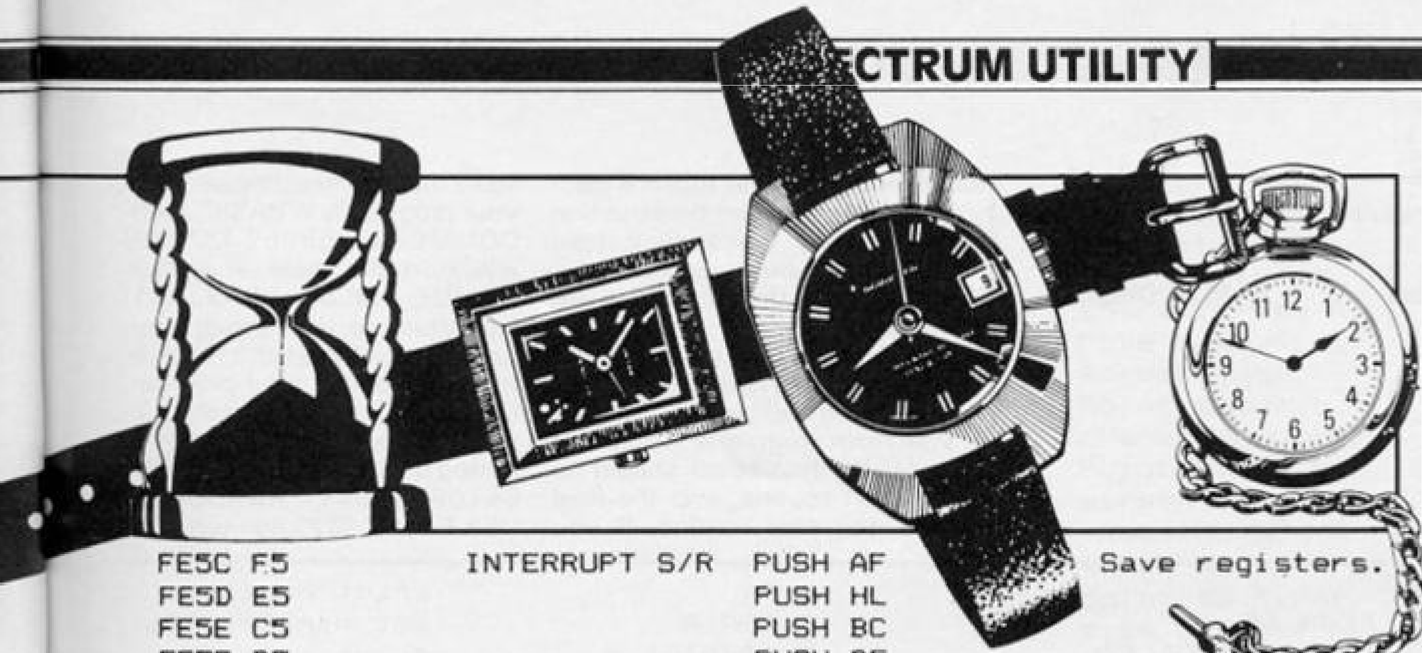
16K users should substitute 7 for the initial F in the adresses, and use the bytes in brackets in place of those underlined in the listing.

```
ORG FE21. (ORG 7E21)

ADDRESS HEX CODE  LABEL     OPCODE          NOTES

FE21 3EFE(28)     START     LD A,FE(28)     High byte of
                                            vector address
FE23 ED47                   LD I,A          to I register.
FE25 ED5E                   IM2             Select
                                            interrupt mode
                                            2.
FE27 FD360200               LD(TV FLAG),00  Print to upper
                                            screen.
FE2B 181C                   JR PR MESS      By-pass message
                                            bytes.
FE2D 16001A5449             DEFM            Print AT 1,24
FE32 4D45160118             DEFM            "TIME"
FE37 30303A3030             DEFM            AT 2,24
FE3C 3A30301602             DEFM            "00:00:00"
FE41 18454C4150             DEFM            AT 3,24
FE46 534544                 DEFM            "ELAPSED".
FE49 212DFE(7E)   PR MESS   LD HL,DEFM 1    First byte of
                                            message.
FE4C 061C                   LD B ,1C        Counter of
                                            number of bytes
                                            in message.
FE4E 7E           RPT       LD A,(HL)       Code of
                                            character to
                                            print.
FE4F D7                     RST 10          Print character
FE50 23                     INC HL          Point to next
                                            message byte.
FE51 05                     DEC B           Decrement
                                            counter.
FE52 20FA                   JRNZ RPT        Jump back
                                            unless printing
                                            complete,
                                            otherwise
FE49 C9                     RET             Return.

FE55 ED56         STOP      IM1             Restore normal
                                            interrupt
                                            status.
FE57 3E3F                   LD A,3F         Restore I
                                            register to
FE59 ED47                   LD I,A          normal value 3F
FE5B C9                     RET             Return.
```



```
FE5C F5          INTERRUPT S/R   PUSH AF       Save registers.
FE5D E5                          PUSH HL
FE5E C5                          PUSH BC
FE5F D5                          PUSH DE
FE60 AF                          XOR A         Reset carry
                                               flag.
FE61 3A01FF(7F)                  LD A,(INT)    Update
FE64 3C                          INC A         interrupt
FE65 3201FF(7F)                  LD(INT),A     counter and
                                               store it.
FE68 FE32                        CP32          Has counter
                                               reached 50d?
FE6A DA4A00                      JPC 004A      If not, exit
                                               via keyboard
                                               check in normal
                                               interrupt
                                               routine.
FE6D AF                          XOR A         Set interrupt
FE6E 3201FF(7F)                  LD(INT),A     counter at 0.
FE71 3A09FF(7F)                  LD A,(SD)     Update seconds
FE74 3C                          INC A         digit counter
FE75 3209FF(7F)                  LD(SD),A      and store it.
FE78 FE0A                        CP0A          Has counter
                                               reached 10d?
FE7A 3847                        JRC PRINT     If not exit
                                               via print
                                               routine.
FE7C AF                          XOR A         Set seconds
FE7D 3209FF(7F)                  LD(SD),A      digit counter
                                               at 0 and store
FE80 3A08FF(7F)                  LD A,(S10)    Update seconds
FE83 3C                          INC A         counter and
FE84 3208FF(7F)                  LD(S10),A     store it.
FE87 FE06                        CP06          Has counter
                                               reached 6?
FE89 3838                        JRC PRINT     If not, exit
                                               via print
                                               routine
FE8B AF                          XOR A         Set seconds 10s
FE8C 3208FF(7F)                  LD(S10),A     to 0 and store.
FE8F 3A06FF(7F)                  LD A,(MD)     Update minutes
FE92 3C                          INC A         digits counter
FE93 3206FF(7F)                  LD(MD),A      and store.
FE96 FE0A                        CP0A          Has counter
                                               reached 10d?
FE98 3829                        JRC PRINT     If not, exit
                                               via print
                                               routine.
FE9A AF                          XOR A         Set minutes
FE9B 3206FF(7F)                  LD(MD),A      digit counter
                                               at 0 and store.
FE9E 3A05FF(7F)                  LD A,(M10)    Update minutes
FEA1 3C                          INC A         10s counter
FEA2 3205FF(7F)                  LD(M10),A     and store.
FEA5 FE06                        CP06          Has counter
                                               reached 6?
FEA7 381A                        JRC PRINT     If not, exit
```

form around 3000 instructions in 1/50sec., it is easy to overrun the time. You will still need to scan the keyboard, and the scanning routine itself is quite a long one. When planning this program I tried to use a machine code version of the handbook BASIC program, using the calculator stack for the calculations, and the ordinary RST10 printing routine for displaying the clock. It crashed. Each calculator stack instruction calls up various ROM subroutines, and the total number of instructions was far too large for the time available. So be careful of ROM subroutines. A one byte instruction on paper can call up a very complex network of instructions in practice.

So, I simplified the subroutine, no longer updating FRAMES, but introducing our own clock counters, the program variables at FF01-FF09 (7F01-7F09 for 16K). This eliminated the calculator stack, but I was still using a series of RST10 instructions for the display, and, I found another snag. Using RST10 alters some of the system variables — TV FLAG, S POSN, etc. So although the subroutine could just be performed in the time, the main program went haywire because of the corrupted variables. When I tried to save all the variables involved at the beginning of the subroutine, and restore them at the end, we were over time again. So out went RST10 in favour of direct POKEs into the screen for displaying the clock.

It is essential to save all the registers at the start of an interrupt subroutine and restore them before returning from it to avoid their corruption causing the main program to crash. Even a BASIC program will crash if you neglect this. Although the clock routine does not appear to end with a series of POP instructions, they are there. It ends by jumping into the usual interrupt subroutine at the point where it scans the keyboard, and the POPs are the final instructions of that subroutine.

The other major difficulty is the devious route the Spectrum takes to its interrupt subroutine when in IM2. On an interrupt, it jumps to an address whose low byte is FFh, and whose high byte it gets from the I register. (Not to be confused with the IX or IY registers). At this vector address, it expects to find the actual subroutine address. So before the IM2 instruction, you must load the I register with the high byte of the vector address, ▶

Figure 1. Available addresses for the 16K user.

ROM ADDRESS. (Vector)	Bytes There	RAM ADDRESS
06FF	DD 71	71DD
0FFF	18 6D	6D18
14FF	69 64	6469
19FF	22 5D	5D22
1EFF	CD 67	67CD
28FF	5C 7E	7E5C

and the vector address must hold the two bytes of the subroutine address, in the usual format of low byte first. In this program, the 48K subroutine is at FE5C, and the vector address at FEFF, so the I register holds FE, and at FEFF and FF00 are the bytes 5C FE. When returning to IM1, the I register must be restored to its usual value of 3F.

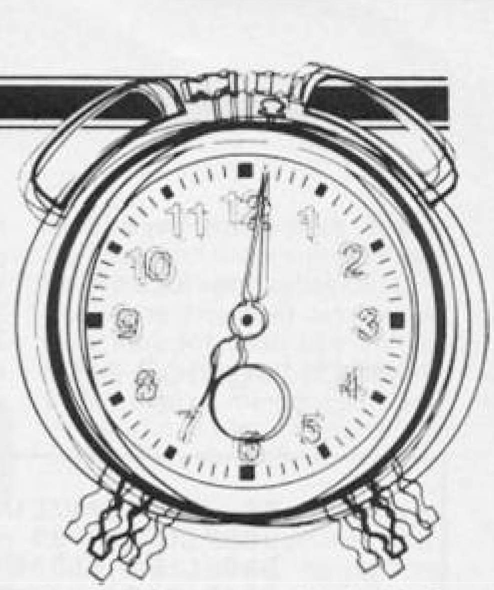
But, there is another complication. If the I register holds any value between 40h and 7Fh, nasty things happen to the screen display. You need someone who knows more about what goes on under the Spectrum's bonnet than I do to explain why (try the editor), but it is so. This does not matter to the 48K user, who can use 80FF or higher, but for 16K folk it means that there is nowhere in the RAM where you can put the vector address. So the 16K user wishing to use IM2 must rummage about in the ROM, looking for an address, whose low byte is FF, at which there are two bytes which point to an address in the 16K RAM. I came up with six. I am not prepared to swear there are no more, but these are enough to go with, (see figure 1). Of these, some are so low in memory that they would leave very little room for BASIC. I stopped looking for others when I found the one at 28FF, which gives the RAM address 7E5C, because the subroutine fits neatly there without overwriting the user-defined graphics.

Conversion

To simplify the conversion between machines, I have placed the 48K routine at FE5C. The listing is for 48K, but I have underlined the bytes which must be changed for 16K, and included the 16K bytes in brackets. By a happy accident, the subroutine ends at FEFE, so that the vector address can follow immediately, between the subroutine and the program variables. If the 'FF' address falls in the middle of the subroutine, the vector address must be placed there, with a jump instruction immediately preceeding it so that the subroutine operation can by-pass the subroutine address bytes.

The start and stop routines do not have to reside in memory where I have placed them. If your main program is in machine code, the first bytes should be the start routine, and the final ones the stop routine. If you leave them where they are when your program is in BASIC, RANDOMIZE USE 65057 (32289) will start the clock and RANDOMIZE USR 65109 (32341) will return to the normal interrupt status. If you wish to pause the clock during your program, i.e. to allow for reading instructions without loss of time, simply stop it with RANDOMIZE USR 65109 (32341). RANDOMIZE USR 65057 (32289) will start

```
                                                   print routine.
FEA9  AF                        XOR A              Set minutes 10s
FEAA  3205FF(7F)                LD(M10),A          counter to 0
                                                   and store.
FEAD  3A03FF(7F)                LD A,(HD)          Update hours
FEB0  3C                        INC A              digit counter
FEB1  3203FF(7F)                LD(HD),A           and store.
FEB4  FE0A                      CP0A               Has counter
                                                   reached 10d?
FEB6  380B                      JRC PRINT          If not, exit
                                                   via print
                                                   routine.
FEB8  AF                        XOR A              Set hours
FEB9  3203FF(7F)                LD(HD),A           digits counter
                                                   to 0 and store
FEBC  3A02FF(7F)                LD A,(H10)         Update hours
FEBF  3C                        INC A              10s counter
FEC0  3202FF(7F)                LD(H10),A          and store
FEC3  3E38           PRINT      LD A,38            Low byte of
FEC5  320AFF(7F)                LD(SCR LB),A       first print
                                                   position stored
                                                   in variables.
                                                   (High byte is
                                                   always 40h so
                                                   need not be
                                                   stored.)
FEC8  3E02                      LD A,02            Low byte of
FECA  320BFF(7F)                LD(VARS LB),A      first variable
                                                   to print
                                                   address stored
                                                   in variables.
                                                   (High byte is
                                                   always FF (7F))
FECD  4F                        LD C,A             Address of
FECE  06FF(7F)                  LD B,FF (7F)       variable to
                                                   print to BC.
FED0  0A                        LD A,(BC)          Character to
                                                   print to A.
FED1  87                        ADD A,A            Multiply mumber
FED2  87                        ADD A,A            by
FED3  87                        ADD A,A            8.
FED4  4F                        LD C,A             8 * number to
FED5  0600                      LD B,00            print to BC.
FED7  21803D                    LD HL,3D80         Address of
                                                   first byte of
                                                   "0" in ROM
                                                   character
                                                   table.
FEDA  09                        ADD HL,BC          Point HL to
                                                   first byte of
                                                   character to
                                                   print.
FEDB  3A0AFF(7F)                LD A,(SCR LB)      Address of
FEDE  5F                        LD E,A             first byte of
FEDF  3E40                      LD A,40            screen position
FEE1  57                        LD D,A             to DE.
```



FEE2 0608		LD B,08	Counter – no. of bytes in character.
FEE4 7E	BYTE RPT	LD A,(HL)	First byte of character to A.
FEE5 12		LD(DE).A	Poke into first screen byte.
FEE6 23		INC HL	Point to next character byte.
FEE7 14		INC D	Point to next screen byte. (Not INC DE which would point to first byte of next character position).
FEE8 05		DEC B	Have all 8 bytes been poked?
FEE9 20F9		JRNZ BYTE RPT	Jump back if not.
FEEB 3A0AFF(7F)		LD A,(SCR LB)	Point to next
FEEE 3C		INC A	screen position
FEEF 320AFF(7F)		LD(SCR LB),A	and store.
FEF2 3A0BFF(7F)		LD A,(VARS LB)	Point to next
FEF5 3C		INC A	variable to
FEF6 FE0A		CP0A	print and check whether printing complete.
FEF8 2802		JRZ EXIT	Exit if so.
FEFA 18CE		JR PR RPT	Jump back for next character.
FEFC C34A00	EXIT	JP004A	Exit via keyboard scan in normal interrupt routine.
FEFF 5CFE(0000)		DEFB	Subroutine address. (Not used by 16K.)
FF01 00	INT	DEFB .	Interrupt counter.
FF02 00	H10	DEFB	Hours '10s'.
FF03 00	HD	DEFB	Hours digits
FF04 0A	:	DEFB	** SEE BELOW
FF05 00	M10	DEFB	Minutes '10s'
FF06 00	MD	DEFB	Minutes digit
FF07 0A	:	DEFB	** SEE BELOW
FF0B 00	S10	DEFB	Seconds '10s'
FF09 00	SD	DEFB	Seconds digit
FF0A 00	SCR LB	DEFB	Low byte of current screen position.
FF0B 00	VARS LB	DEFB	Low byte of address of variable to print.

** With the exception of FF04 and FF07, the locations FF01 – FF0B are the variables which will be altered as the program proceeds. The CODE of the digits is 30h – 39h, and the CODE of ":" is 3Ah. As the value stored in the variable for each digit is not the CODE, but the true value, the value stored is (CODE-30h). To store ":" in the same convention, it is therefore neccessary to store 0A wherever it is required to print ".".

it again from where it left off. To start it again from zero, you must POKE 0 into the following variables:

Spectrum 48 K

FF02	65282
FF03	65283
FF05	65285
FF06	65286
FF08	65288
FF09	65289
FF01	65281

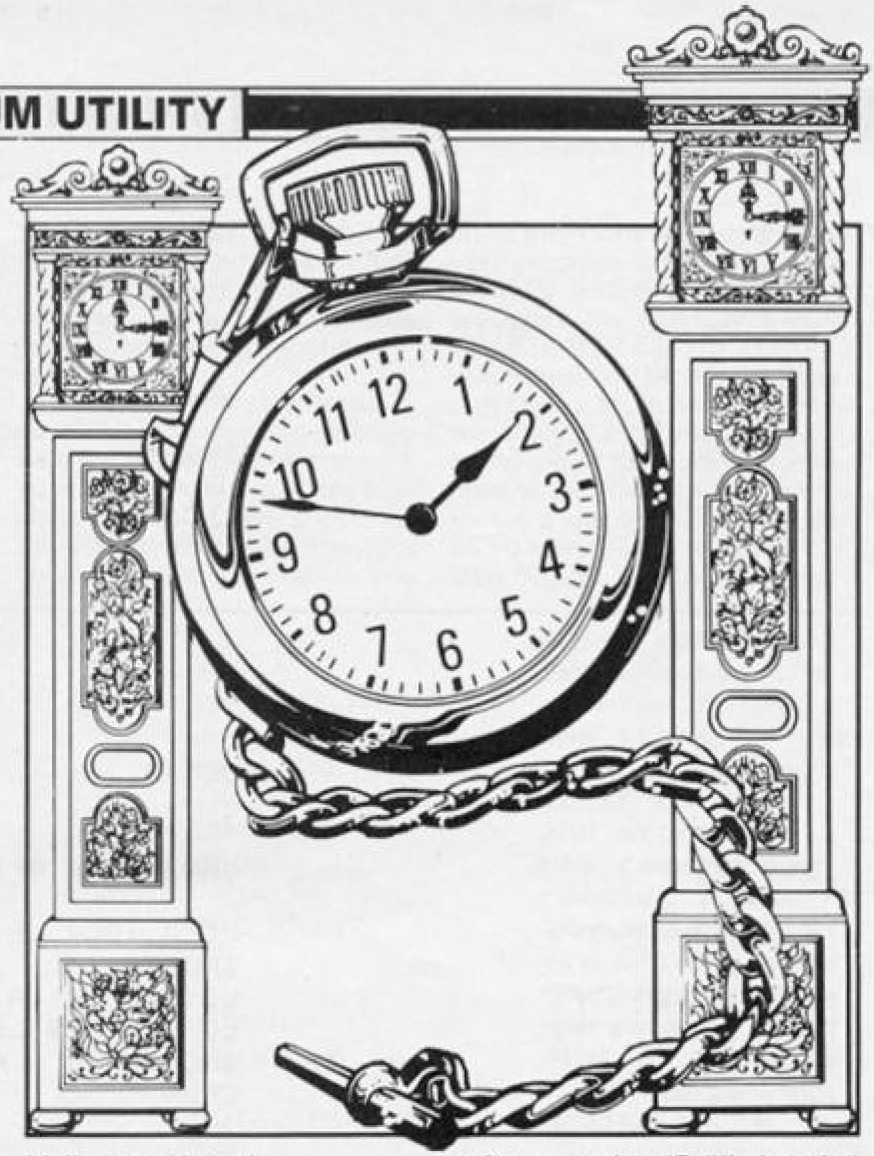
Spectrum 16 K

7F02	32514
7F03	32513
7F05	32517
7F06	32518
7F08	32520
7F09	32521
7F01	32513

Do not overwrite the clock "window" with anything else during your program or you will lose the words "TIME ELAPSED". The clock itself will reappear, however, even after CLS. The window is AT:

0,	24-31
1,	24-31
2,	24-31

Enter CLEAR 65056 (32288) to protect the clock above RAMTOP. If you have just time for an hour's programming before the pub opens or you have to feed the cat, you can run the clock and keep an eye on the time while you work. Once the code is loaded, RANDOMIZE USR 65057 (32289) will start the clock, and you can use the Spectrum quite normally. NEW will delete "TIME ELAPSED", but so long as you remembered the CLEAR 65056 (32289) instruction, the clock will continue. SAVE, LOAD, VERIFY and using the printer will pause it briefly, and a scroll will remove "TIME ELAPSED" and give you a copy of the time the scroll occurred above the running clock, but otherwise it will run accurately for 99hrs., 59min., 59secs., after which the printing will be corrupted. (Anyone who wants to spend more than four days and nights at the key- ▶



board can write their own routine!)

The following BASIC will allow you to insert any commands you need for your own program.

Save the program on tape *before* running the clock, so that it always starts from zero when loaded.

```
10   RANDOMIZE USR65057 (32289)
9899 GO TO 9999
9900 CLEAR65056 (32288)
9910 LOAD "" CODE
9999 RANDOMIZE USR65109 (32341)
```

SAVE "program" LINE 9900: SAVE "clock" CODE 65057 (32289), 235

The clock runs in real time, but you can change this by POKEing other values into FE69, 65129. (7E69, 32361). Values lower than 32H, 50d, will speed the clock up, higher values will slow it, enabling you to change the level of difficulty in your program. If you wish to make some operation in your program conditional on a certain time having passed, you can PEEK the variables. For example, PEEKing M10, FF05, 65285 (7F05, 32517) will tell you that 10 minutes have passed each time the value held there changes.

Sources

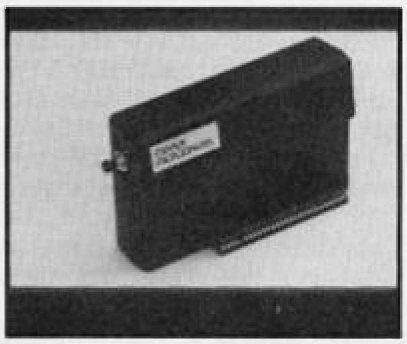
When devising the program, I made extensive use of the following books:

LOGAN.
Understanding your Spectrum. (Technical explanation of interrupt instructions)

LOGAN & O'HARA.
The complete Spectrum ROM disassembly. (Details of the purpose and structure of all ROM subroutines)
(Both above published by Melbourne House)

ROSS-LANGLEY.
The Spectrum machine code reference guide. (Full listing of all ROM addresses and the bytes held there.)
(Published Interface).

Mirage Microdriver



The bane of any games player is the time taken waiting for your favourite game to load. The Sinclair Microdrive was eagerly accepted as a solution to this problem, but no software houses produced games using microdrives (mainly due to the cost of the cartridges). Even those who attempted to convert games from tape onto microdrive found that the time and knowledge required to get round modern anti-piracy methods was too much trouble to bother with.

Now, I am completely opposed to the theft of software, but the majority of us simply want to use our microdrives as a fast means of getting programs up and running. Until recently, microdrives have been a bit of a white elephant from this point of view. The MIRAGE MICRODRIVER, however, is the answer to the gamesman's prayers. About the size of a large joystick interface, it fits onto the port at the rear of the Interface 1, and is compatible with Interface 2 and all the other well known joystick interfaces. In fact, it proved compatible with every peripheral I tried, including printer and Beta disc drive interface!

The unit is supplied with a typed five page 'manual', which explains in simple detail how to operate it. Even if these instructions had not been included, there would have been no problems, since, once you press the button on the side of the unit, there are prompts at every stage of the process.

The unit takes over when this button is pressed, and saves the program by making a complete dump of the RAM. Games can be saved at any point during play, and will continue from that point upon subsequent reloading (great for those long and complex arcade adventures).

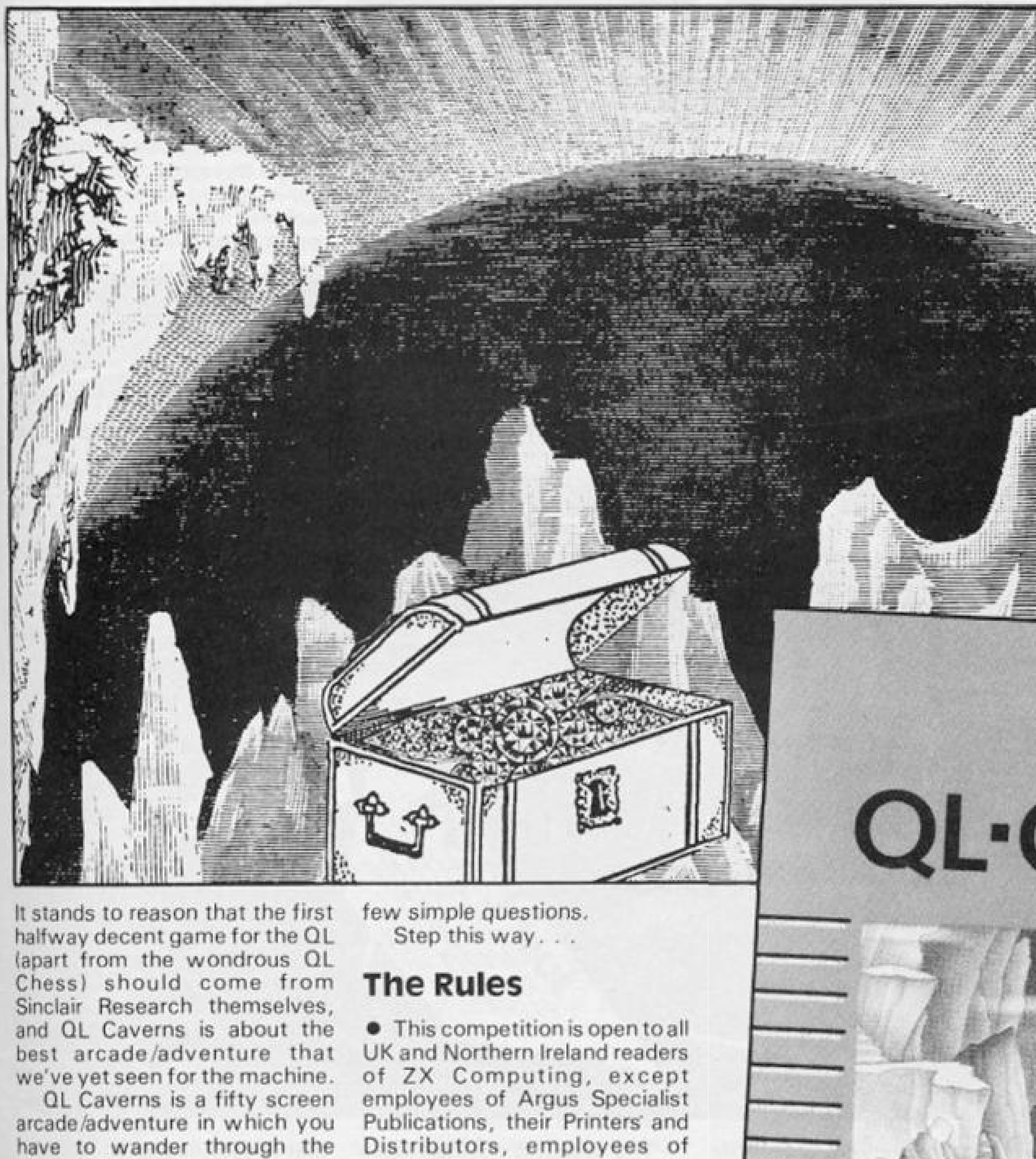
Of course, there has to be one disadvantage, though I consider it to be only a minor one, and this is that any program which contains a built in SAVE or LOAD routine for data still has to perform that function using a tape recorder. However, as most games keep their data in memory, all that is often needed is a simple reSAVE of the whole program, wiping out the old one if it is no longer required.

The chance of misusing the unit for piracy purposes is cut down by the fact that programs copied will only reload if the Microdriver is still present (an excellent idea).

The Microdriver is incredibly simple to operate, was 100% effective on all the programs we tried it with, and is an essential device for all microdrive owners. It ought to have been built into Interface 1 in the first place. They should have called it the Miracle Microdriver!

The Mirage Microdriver costs £44.95, from Mirage Microcomputers Ltd, 24 Bank St, Braintree, Essex CM7 7UL. ■

QL Caverns Competition



It stands to reason that the first halfway decent game for the QL (apart from the wondrous QL Chess) should come from Sinclair Research themselves, and QL Caverns is about the best arcade/adventure that we've yet seen for the machine.

QL Caverns is a fifty screen arcade/adventure in which you have to wander through the caverns in search of an enormous number of diamonds (almost 400). Along the way there are a number of objects, such as parachutes, jetpacks and boats, that you can use to help you in your explorations.

QL Software may not yet be in the Knightlore League, but this is the QL's own version of Jet Set Willy, at the very least. Anyway, we've got half a dozen copies of QL Caverns to give to software starved QL owners, and all you have to do to enter the competition is to answer a few simple questions.

Step this way. . .

The Rules

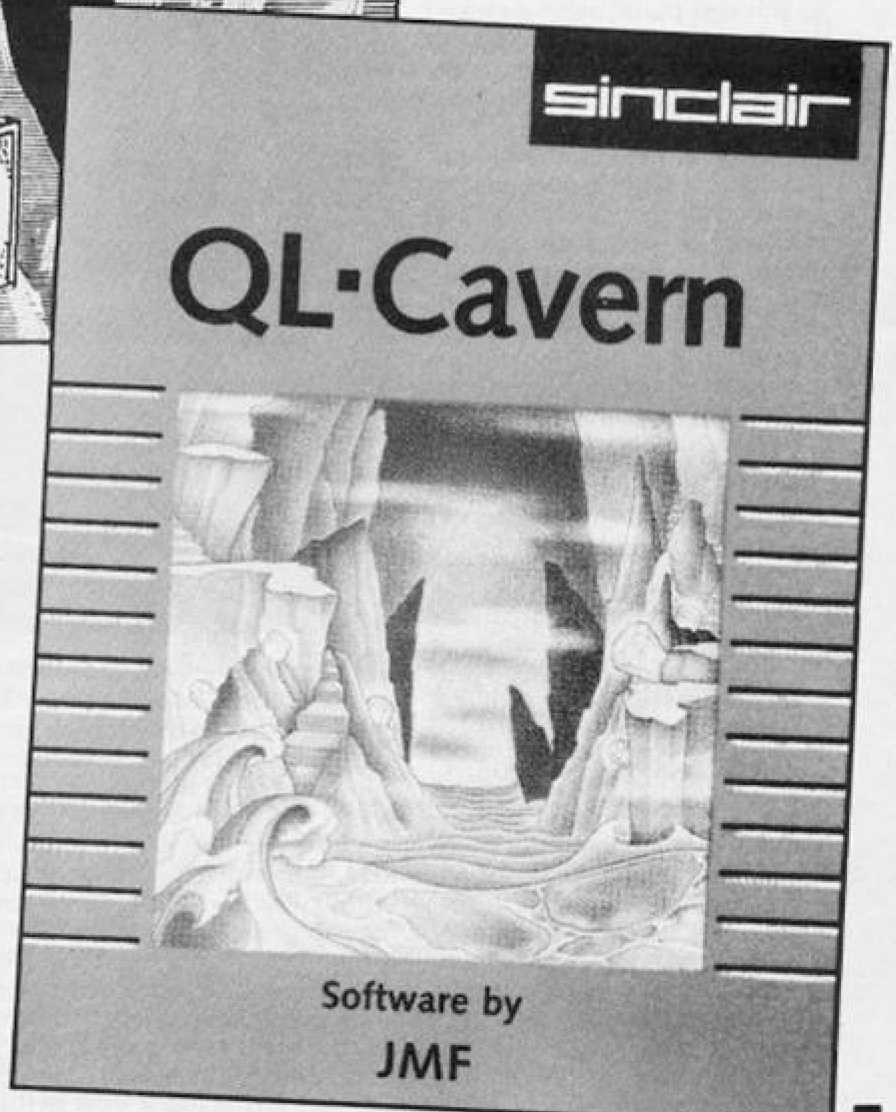
- This competition is open to all UK and Northern Ireland readers of ZX Computing, except employees of Argus Specialist Publications, their Printers and Distributors, employees of Sinclair Research, and anyone else associated with this competition.
- All entries must be postmarked before the 30th November 1985. No correspondence will be entered into with regard to the competition, and it is a condition of entry that the Editor's decision is final.
- As long as each entry is sent on a separate postcard or envelope, then there is no limit to the number of entries that may be submitted by an individual.
- The winners will be notified by post, and the results published in a future edition of ZX Computing.

The Competition

All you have to do is answer the following questions.

1) Name the game, written by Matthew Smith for the Spectrum, that started the platform game craze.
2) Name the game from A'n F Software that featured a character called Henhouse Harry.
3) What was the first game for the QL called (released by Psion/Sinclair)?
4) As a tiebreaker, tell us (in twentyfive words or less) which Spectrum game you would most like to see converted onto the QL, to use the QL's extra memory and graphics capabilities.

When you've worked out all the answers, just write them on the back of a postcard or envelope and send your entries to; QL Caverns, ZX Computing, 1 Golden Square, London W1R 3AB.



Harwood's Hypothesis

Shuffling over from software reviews, David Harwood is now delving into ZX's tottering book pile.

Moving on to better and greater heights I am now positioned on the Book Page, where in each issue a critical and informative view on the current books will be presented.

Purchasing a computer book is totally different to buying a games tape. Whereas many might not think twice when spending £5.95 on an Asteroids shoot-em-up type game, in order to spend £5.95 on a book, the prospective purchaser will go through much mental strain. Will the book gather dust on the shelf or will it, in no time at all, look like a book from the archive section of the local library? Once decided a purchase can be made and it is anybody's guess if the correct decision was made.

Harwood's Hypothesis intends to evaluate each book carefully, so that you, the people who keep the computer business going, can make sure that your money isn't wasted.

Well, enough of my babbling, here's the first book.



Spectrum Logo

This is published by Interface Publications and written by the Master-over-all-computer-books, Tim Hartnell. The book claims to provide you with a program which will turn your ZX Spectrum into a LOGO based machine.

For those of you not in the know, LOGO is a graphics programming language. It was invented for the purpose of introducing children to computers, and allows an imaginary turtle to be controlled as it moves around the screen to produce various designs and pictures. The LOGO program in the book conforms to the LOGO COMPUTER SYSTEMS, INC vocabulary. This enables the experienced user to enter the world of LOGO and one that is compatible with other systems.

The program is well written and is listed in two versions — with and without the REM statements. This is an excellent idea, not only to fill the book up with a few extra pages (!) but also as the total novice to computing only wants to type in the necessary lines and not bother with pointless keying in. For the more experienced programmer, the REM's are very well detailed and give an excellent view of the program's action.

The book is written in a friendly, easy to understand way. From the introduction, right through to the many examples, the reader is taken at a steady pace through the learning process, starting with the basics of the language and then progressing, so that by the end of the book an in depth knowledge of LOGO can be achieved.

Included in the book are numerous LOGO examples, from a six-sided figure (called a sexygon by Tim Hartnell) to lots of tiny SIR CLIVE's, holding hands in a circle.

The book is a bargain at £2.99, even for the experienced reader who finds Tim's teaching a bit simple, as the LOGO emulation is very good. My only grumble being the speed at which LOGO runs, but this can only be blamed on Sinclair and not the author.

The On-Line Computer Handbook

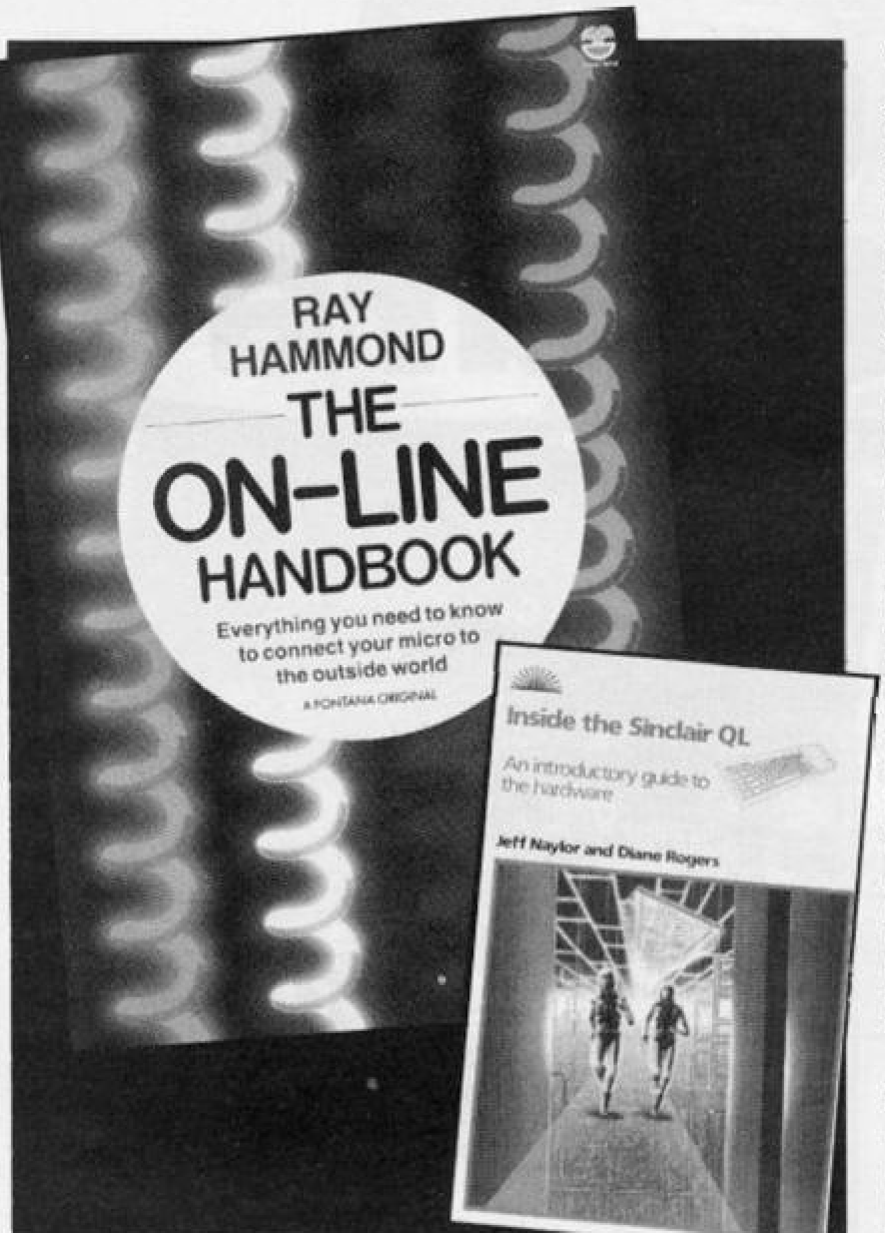
Written by Ray Hammond and published by Fontana, this book claims to supply all the information required to connect your humble micro to the outside world, and I can honestly say that it does live up to its claim. When I received the bundle of books, I first of all had a quick thumb through all of them. On picking this one up, much like a good thriller, I just could not put it down.

The reader is initially introduced to what is meant by the term 'going ON-LINE' and how easy this is. The advantages of doing so are also heavily emphasised (perhaps the author has the controlling interest in Telecom shares!). The book then progresses to show what is needed, whether using a ZX80

or an IBM PC to go on-line. Various technicalities are explained, from the basics of the ASCII character set to the modem and interface operation and, if armed with a soldering iron, even how to wire up an RS232 cable (WOW!). The explanation of how data is sent across a telephone line is also included and is very interesting.

The remainder of the book then shows how to log on to various computer Bulletin boards, Prestel, Micronet and various useful Databases. Some of the above methods need passwords and up-front payment, and the reader is fully instructed on how to go about this. Many examples of the author's on-line experience, showing what happens when you actually go on-line are also given.

This book is a must for anyone who has the slightest interest in communications and at the mere price of £4.95, it is a great buy. This book will be taking pride of place on my bookshelf for quite a time. One last note — the book does not tread onto the path of the so-called 'hacker' and is totally legitimate.



Inside The Sinclair QL

Sunshine have produced a book for those who 'want to know how their Sinclair QL ticks.' INSIDE THE SINCLAIR QL is written by Jeff Naylor and Diane Rogers and provides an introductory guide to computer hardware.

The book is split into two sections. The first explains basically what a computer is, and the second strips the QL down. The basis of computing is at first given, starting off with simple circuit diagrams and progressing on to how a computer makes sounds and the idea behind the video picture (or VDU). Then a breakdown of the components of a computer is given, from the CPU to the ROM and RAM. This section was informative and fairly comprehensive but would not, I feel, appeal to the early beginner.

The second section which, incidentally, is as comprehensive as the first, starts with a brief advertising blurb for the QL, showing its family tree (ie ZX80, ZX81, ZX SPECTRUM . . . SINCLAIR QUANTUM LEAP) and the QL specification. The QL hardware structure is then described, along with the strip down of the Motorola 68008 microprocessor. A whole chapter is devoted to the 68008 instruction set as well as a chapter on exploring machine code on the QL.

The final chapters, explaining the memory, video display and INPUT/OUTPUT methods have, perhaps been printed in the incorrect place and should have appeared at the same time as the QL hardware was described and not after the chapter on machine code.

My conclusion is that this is quite an informative book, not at all intended for the beginner but someone who perhaps knows his/her BASIC, or SuperBASIC and wants to learn more about the computer. The book, retailing at £6.95, does cover the vast area of technicalities on the Sinclair QL and is well researched.

Melbourne House

The last two books are machine code based and directed at the experienced programmer or perhaps very enthusiastic beginner. Both are published by Melbourne House, the former costing £8.95 and the latter £9.95.

Spectrum Shadow Rom Disassembly, by Gianluca Carri, is written in two parts. The first shows how to extend Sinclair BASIC to include various useful commands, random file handling on the microdrive, extended BEEP and EDIT commands. Full details of the implementation are given along with very clear explanations.

It is possible to add extra commands to the Spectrum with an Interface 1 unit, containing the Shadow ROM, because when an invalid command is found, a search is made in the existing Spectrum ROM. If this command is not present, then the Shadow ROM is searched. If the command is not found again, then it is at this point that we can jump to our own routine, and carry out the extra commands.

There are two types of Shadow ROMs, and this book accommodates both. There might even be another release of the Shadow ROM software, but the author still explains fully what to do if this is the case.

The second part of the book gives a full disassembly of the Shadow ROM in the Interface 1, and how to link up and use these various routines. This is ideal literature for the Z80 programmer wishing to either add commands to BASIC, access the microdrives via machine code or see what's in the Shadow ROM.

The second new book from Melbourne House is the **Z80 Reference Guide** by Alan Tully, which is intended as a reference manual for the Z80 programmer.

The book is designed to help the Z80 programmer, who perhaps is stuck with a programming problem, to choose the correct Z80 commands to use. An excellent chapter is included which lists the Z80 instruction set and where best to use the various commands. The bulk of the book is a summary of the Z80 instruction set, which gives full details of each individual instruction, along with the effect on the flag register and the timing of the specific instruction. Excellent examples of usage are also given for all but the most simple instructions.

The book covers a vast topic and is ideal for anyone interested in machine code, either to be used by itself or as a supplement to an existing 'Learn How To Program' guide. Although this book is expensive at £9.95, it is well worth having an empty pocket for all those who are into machine code (*better still, you could rush out and enter our competition for copies of this book, elsewhere in this issue — Ed*).

FOOTNOTE

It appears that machine code books are more expensive than the BASIC book. This perhaps is due to the amount of time needed to produce a masterpiece. I wonder what would happen if the price was reduced? Would more copies be sold? A happy medium needs to be achieved, but is it at £9.95?

Interface Publications, 9-11 Kensington High St, London W8.

Fontana Books, 8 Grafton St, London W1.

Sunshine, 12-13 Little Newport St, London WC2.

Melbourne House (Publishers), Castle Yard House, Castle Yard, Richmond.

Simulation Software

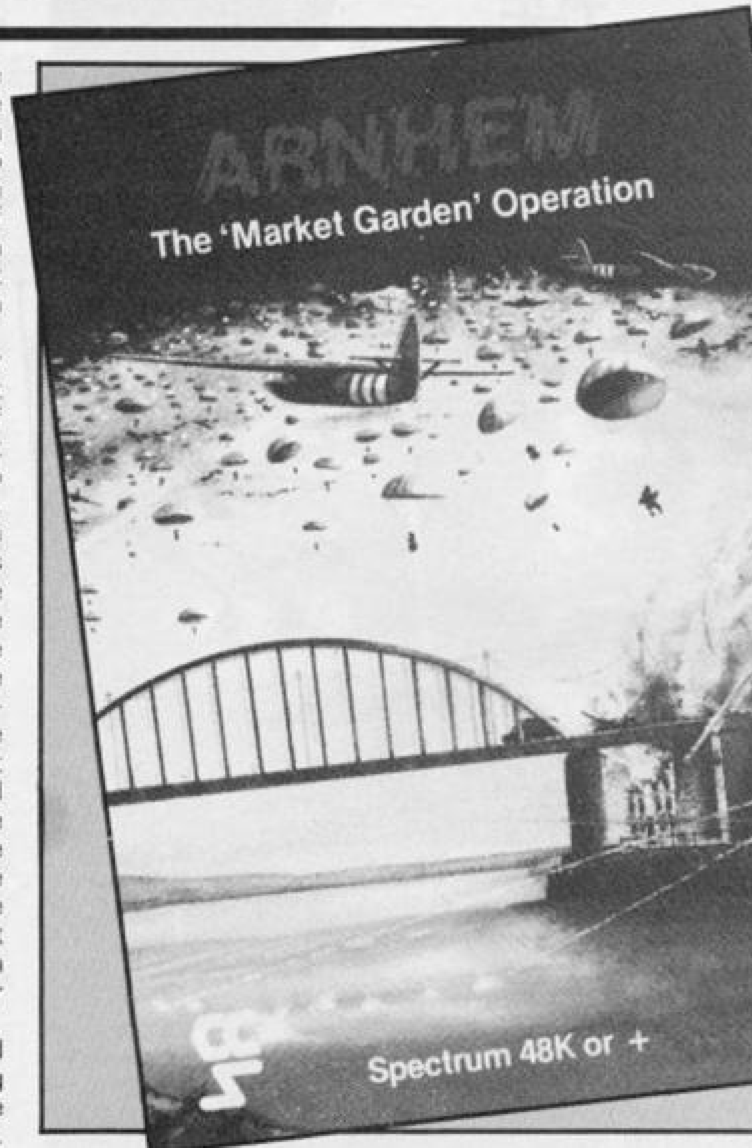
Adventure games can be generally likened to a puzzle which has to be unravelled in a certain order or sequence to complete it. I know there are a couple of notable programs of this genre which have more than one way of solving them but these tend to be the exception rather than the rule.

Arcade games can generally be summed up as requiring fast reflexes and usually consist of shooting, chasing or dodging. It is here that the biggest overlap with our chosen feature exists. Many arcade games now have a strategic element and it is almost impossible to separate the two styles. However, we have tried to categorise a large number of the programs on the market and made our own decision as to which group we were going to allocate them. The fact that a game has been omitted does not mean it doesn't have some strategic elements and we recognise that many of the latest generation of arcade games need as much (if not more) of a strategic approach than many we included in our list!

This feature is entitled Wargames but we are including most of the games which require thought, strategy and luck. Unlike adventure games, to win you have to get many factors in your favour, balance between a multitude of options, and it helps to be a little bit lucky. Also, there is usually more than one way to win.

At the heart of any good program of this type is an algorithm which takes into account all of the relevant factors, balances the importance of each (and this may change at various stages of the game) adds just enough — not too much — of a random, chance factor and presents the results of that turn.

In board games in particular the algorithm may be replaced by a selection of options included by the programmer and the computer needs to pick the most appropriate responses to the player's moves.



Many moons ago when I was writing/converting programs for the ZX81 I came across the first of this kind of game, called Stockmarket and sold by ASP for the BBC and other computers. I made the conversion (which I'll put in the mag one day) and I have been playing the game ever since!

Wargames

Probably the most complex of the strategy programs, these range from recreating actual military scenarios as in ARNHEM, THE BULGE, BATTLE FOR MIDWAY, etc to imaginary situations which could be set in the past, present, or future, such as VIKING RAIDERS, NATO ALERT or REBELSTAR RAIDERS.

Lothlorian are probably the company most associated with this style of game and have made great strides over the past few years, their CONFRONTATION along with Red Shift's APOCALYPSE are probably the most well known and successful games so far. These are both fairly traditional and tend to operate essentially in 'move time' and are predominantly text based. The advantage of this is that you can go away and think about your next move, saving the game at various stages as you go.

Although this appeals to the deep thinker, it lacks realism in so far as you couldn't wander off in the middle of, say, the battle of Waterloo, and come back a few days later to make your next attack. If that has been the case, Napoleon would have been overjoyed, and we would probably all be speaking French!

The first game to impress me with real time continuous action was STONKERS by Imagine (the original company) and this is still available through Beau Jolly. I have not yet managed to beat it, but then, he who fights and runs away . . .

The new generation of programs seems to be following this continuous action idea and I have also been beaten at the Bulge and at Midway. You really need more time to get to grips with them than this job allows (that's my excuse and I'm sticking to it!).

Business and Management games

CCS, Cases Computer Simulations, pioneered this type of program and have brought them to a tremendous degree of sophistication. Their earlier programs tended to be a little too random but the latest releases provide a plethora of options.

In this category is one of the most consistent best sellers ever written for the Spectrum or ZX81 or most of the other main machines. Kevin Toms' FOOTBALL MANAGER, marketed by Addictive Games. Over the years it has developed from ZX81 text only to an all singing, dancing, graphics wonder. A teacher I know considers it the only thing capable of keeping an awkward class engrossed.

There are many variations on this theme, Addictive themselves are about to produce one or two, in the meanwhile AMERICAN FOOTBALL, NEW WHEELS JOHN and FORMULA ONE are favourites of mine.



Strategy

All the programs in this feature depend on a strategy element of some kind, the ones in this particular category are oddments in a way, they didn't seem to quite fit in any other section.

ARCHON and CHAOS are almost but not quite board games and keep me occupied for many hours, and one day I'm going to find the time to give ALIEN, SHADOWFIRE AND BROAD ST. the attention they deserve.

Graphic Sports Simulations

This section gave us our biggest discussion as most really fall into the arcade games category with success depending on the strength of your wrist and the durability of your joystick.

Still, a few, in particular the Golf games — always a favourite with ZXC readers when we publish one — have a great deal of strategy involved.

We include as full a list as we could get, and suggest you check them out at a shop if you are in doubt as to whether they meet your requirements. Personal favourite — THE FOREST, a graphic orienteering simulation from Phipps Associates.

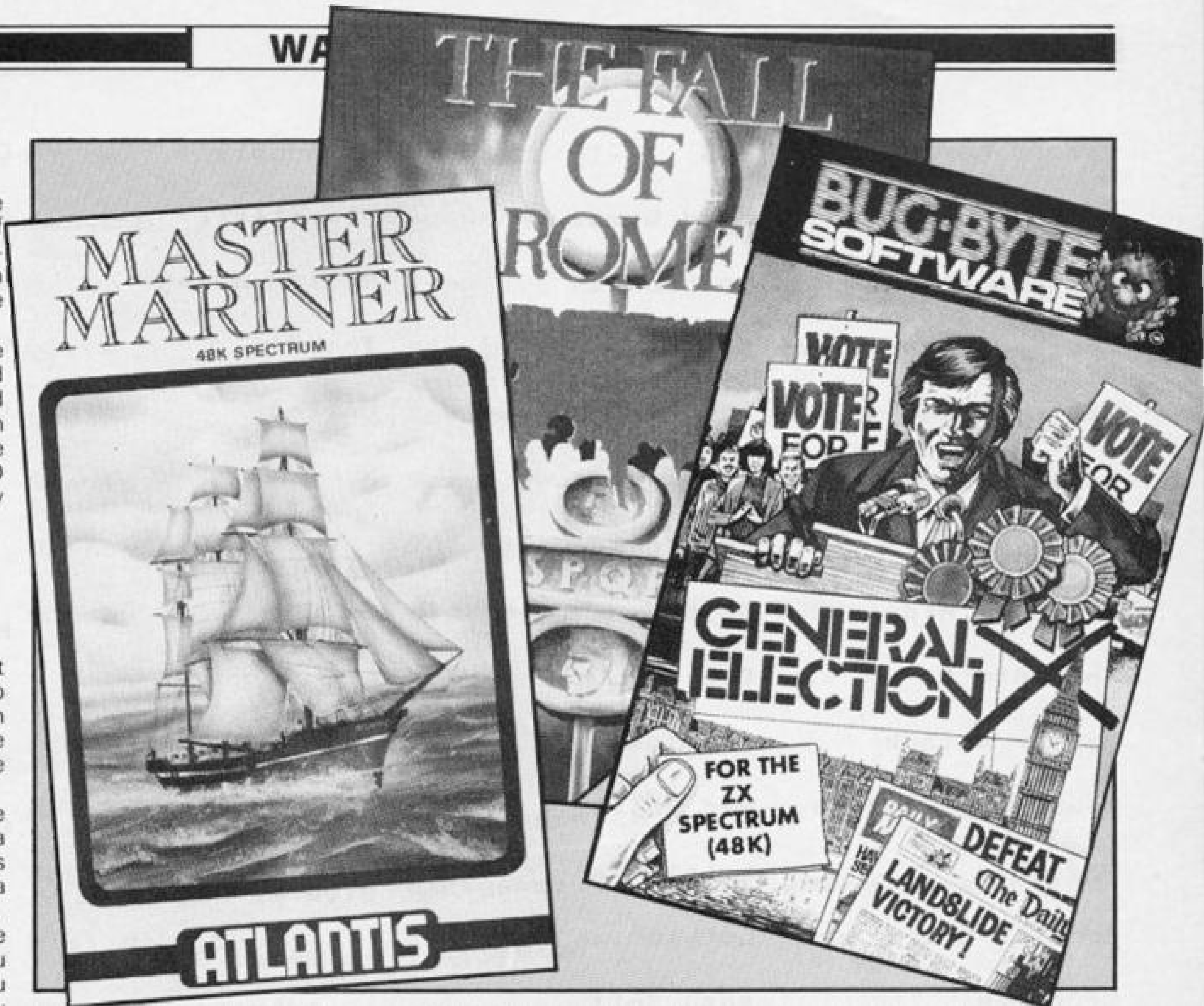
Other Simulations

Flight Simulations have been popular for as long as computers themselves. One of the first acclaimed programs for the ZX81 was Psion/Sinclair's FLIGHT SIMULATION which was rapidly converted for the Spectrum.

Hewson Consultants produced their version which operated impressively in 16K and then took off on a different tack with their Air Traffic Controller series. They have come down to earth with their latest and quite brilliant simulation, SOUTHERN BELLE. Great for us ageing computists who can actually remember steam trains. I have never been able to master any of the flight simulation programs (I have only five digits on each hand) but Belle has enough simple options that I actually succeeded.

Traditional and Board Games

This is another type of program which has found favour with ZXC readers. The race to produce the best CHESS program was another of the earliest endeavours. In my own collection I am proud to own the amazing 1K ZX81 program. It may not be sophisticated but it's a masterpiece of programming.



There is much controversy about who has produced the best chess program so far. Each has different strengths and features and we have printed whole articles comparing just two. Without sticking our necks out, MASTERCHESS, CYRUS IS CHESS and SUPERCHESS 3.5 have a large number of supporters. It is worth mentioning the graphically amazing QL CHESS which I saw at a microfair, the company haven't sent us a review copy so I can't comment on its playing ability but it looks superb.

Board game conversions tend not to be my cup of tea, I do see the value of using the computer as an opponent when a human one isn't available, but for me the joy of playing is the contact with other people. Having said that I must own up to an occasional sneaky game of Sinclair/Psion's SCRABBLE and BACKGAMMON.

Endgame

So there we are, it seems that whatever our most secret desires, be it changing the course of history, making a million, flying a plane or driving a train, there is a way in which the humble computer can pander to those dreams.

There has been some comment at times over the undesirability of bloodthirsty games. Automata have taken a stand and claim there is no overt violence in their programs, a bit of smut maybe, but no violence. My personal view is that provided there is no gratuitous, tasteless graphic violence and as long as over everything there is the realisation that it is just a game, no harm is being done.

I know there are arguments both for and against this point and I have no intention of starting a long discussion on the topic. Quite simply I hope you enjoy this feature and I apologise if I have given offence to anyone.

Software Roundup

Within this broad confines of this feature I have listed as many programs as I could find under general headings. As I have said before, there are many programs which are hard to categorise and so we have exercised our opinion and omitted any which we believe to be primarily arcade games.

Our apologies if we left out any which you or the company think should have been included, this may not have been because we considered it to be arcade but because we hadn't heard of it!

If we have omitted any which you think we should have included then drop us a line and we'll put the record straight.

WARGAMES

Air Defence	CCS
Apocalypse	Red Shift
Arnhem	CCS
Atram	PD Visual Marketing
Battle 1917	CCS

WARGAMES

Battle For Midway	PSS
Bulge	Mind Games
Confrontation	Lothlorian
Doomdark's Revenge	Beyond
Fall of Rome	APS
Insurgency	CCS
Johhny Reb	Lothlorian
Lords of Midnight	Beyond
Nato Alert	CCS
Panzer Attack	Lothlorian
Rebelstar Raiders	Red Shift
Redcoats	Lothlorian
Special Operations	Lothlorian
Stonkers	Beau Jolly
Viking Raiders	Firebird
War Zone	CCS
War 70	CCS

BUSINESS/MANAGEMENT SIMULATION

Airline	CCS
American Football	Mind Games
Auto Chef	CCS
Brewery	CCS
Football Manager	Addictive Games
Formula One	CRL
Grand Prix Manager	Silicon Joy
Millionaire	Incentive
Mugsy	Melbourne House
New Wheels John ?	Automata
Oligopoly	CCS
Planetfall	APS
Racing Manager	Virgin
The Biz	Virgin
United	CCS

STRATEGY

Alien	Mind Games
Archon	Ariolasoft
Blue Riband	CCS
Chaos	Games Workshop
Give my Regards to Broad Street	Mind Games
Shadowfire	Beyond
Stockmarket	CCS
Superpower	CCS

GRAPHIC SPORTS SIMULATIONS

Brian Jacks Superstar Challenge	Martech
Chequered Flag	Sinclair
Daley Thompson's Decathlon	Ocean
Full Throttle	Micromega
Frank Bruno's Boxing	Elite
Jonah Barrington's Squash	New Generation
Golf	Virgin

Grand National	Elite
Hyper Sports	Imagine
International Basketball	Elite
Match Day	Ocean
Match Point	Sinclair
Moortown	Hornby
New Berkdale	Hornby
Nick Faldo's Open	Mind Games
One on One	Ariolasoft
Olympics	CRL
Ralley Driver	Hill Macgibbon
Rocco	Gremlin Graphics
Royal Berkdale	Ocean
Run for Gold	Hill Macgibbon
Ski Star 2000	Richard Shepherd
Steve Davis Snooker	CDS
The Forest	Phipps
The Open	CCS
Troon	Hornby
Video Pool	OCP
World Series Baseball	Imagine
Yacht Race	Hill Macgibbon

OTHER SIMULATIONS

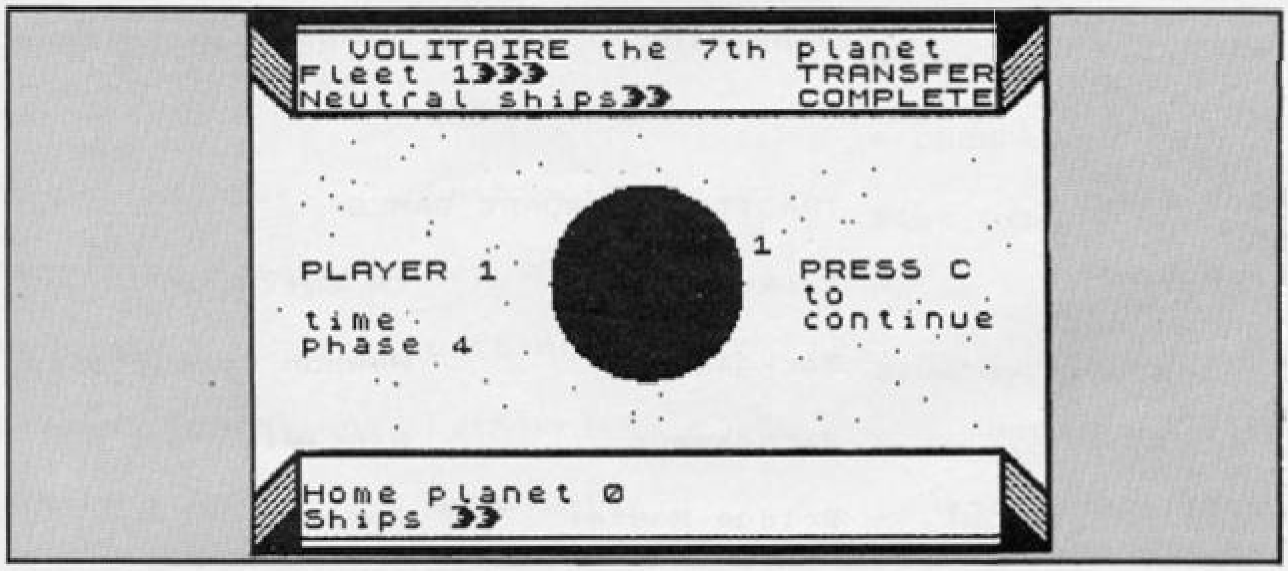
Fighter Pilot	Digital Integration
Flight Simulation	Sinclair
Heathrow/ATC	Hewson Consultants
Juggernaut	CRL
Jump Jet	Anirog
Night Flight	Hewson Consultants
Southern Belle	Hewson Consultants
Space Shuttle	Activision
747 Flight Simulator	DACC

TRADITIONAL/BOARD GAMES

Backgammon	CP Software
Backgammon	Hewson Consultants
Backgammon	Sinclair
Bridge Master	Serin
Bridge Player 2	CP Software
Cluedo	Leisure Genius
Cyrus IS Chess	Sinclair
Draughts	CP Software
Masterchess	Mikro Gen
Maverick	CCS
Mind Games	Oasis
Monopoly	Leisure Genius
Reversi	CP Software
Scrabble	Sinclair
Superchess 3.5	CP Software
The Turk (chess)	OCP
Voice Chess	Artic
Yakzee	Automata
Yatzee	CP Software

Sappro

A two player inter-planetary strategy game by G. Harris from the planet Leicester.



Sappro is a planetary system consisting of ten planets. You are the ruler of one of the planets and your opponent the ruler of another. The object of the game is to capture your opponent's home planet before he captures yours.

Somewhere in the system there is a third planet which is inhabited by pacifists, this planet is protected by an impenetrable defence screen making it impossible to capture. In order to capture your opponent's home planet you must be able to reduce the number of enemy space ships protecting their home planet to zero for two consecutive turns.

Space ships are produced by your home planet, your enemy's home planet and the neutral planet, at the end of each time phase. Ships can be transferred from your home planet and the neutral planet to your fleet. Transferring ships from fleets to planets is not possible.

Players should enter their moves individually while their opponent turns away from the screen, or leaves the room. Unless this secrecy is maintained there is little point in playing the game. Prompts for the change of player/end of turn are given to allow this to take place.

Playing the Game

The first screen is an introductory page and from now on all moves should be made in secret. The second screen displays the names of the ten planets in the system. Each player should now enter his choice of home planet without the other player seeing. When the home planets have been chosen the main display is drawn and the prompt "Player 1 Press Enter" is shown. Player two should now make his exit.

LOOP

When the player has pressed ENTER, hie personal information is displayed with the number of ships in his fleet shown at the top and his planet and the number of guarding ships at the bottom.

Player one now chooses which planet he wants to send his fleet to, types the planet number and presses ENTER. The chosen planet is now drawn in the centre of the screen, its name displayed and any information regarding enemy ships given. If there are enemy ships then combat takes place, this is totally computerised as human decisions are too slow in this age of high technology warfare. The results are displayed at the top of the screen.

Transfer of ships can take place from either your home planet or the neutral planet, to your fleet. If it is possible to transfer ships then you will be prompted for an input of the number you wish to transfer. The sequence is now repeated for player two. Player one should depart and player two should follow the instructions from the label LOOP.

```
   1 REM ***********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         ***********************

   5 DIM p$(10,12): FOR i=1 TO 1
0: READ p$(i): NEXT i
   6 DATA "HOTENTOT","JAKURUTI",
"NUMERATOR","DARTH","AGAR-AGAR",
"LAPIS LAZULI","VOLITAIRE","GEOD
ES","IGNUS FATUUS","CRYOCLOS"
  10 BORDER 0: PAPER 0: CLS
  11 INK 4
  15 GO SUB 3000
  17 GO SUB 7000
  20 LET cs=0: LET trn=0
  21 LET bb=0
  25 LET o$="0"
  30 LET g=0: LET d2=0
  35 LET sc=0: LET end=0
  40 LET n1=0
  45 LET ed2=0
  50 LET tr=0
  55 LET b2=0
  60 LET d=0
  63 LET a$="0005"
  65 LET b$="0005"
  70 LET a=0
  73 LET f$="0000"
  75 LET e$="0000"
  80 LET c$="0000"
  83 DIM s$(32)
  85 GO SUB 600
```

```
 87 GO SUB 6000
 90 LET tn=0: LET b=0
 100 BORDER 0: INK 7
 105 LET in=0: GO SUB 9000
 110 PRINT AT 12,2;"time";AT 13,
2;"phase ";trn
 120 PRINT  PAPER 2; INK 0;AT 10
,2;"PLAYER 1"
 121 GO SUB 1300
 122 IF end=1 AND b$(1)=c$(1) TH
EN  GO SUB 4000
 130 PRINT  INK 2;AT 2,2;"Fleet
1"
 135 FOR n=1 TO VAL a$(3 TO 4)
 136 PRINT  INK 2;AT 2,8+n;"A"
 137 NEXT n
 150 PRINT : PRINT  INK 4;AT 19,
2;"Home planet ";c$(1): PRINT  I
NK 4;AT 20,2;"Ships": FOR n=1 TO
 VAL c$(3 TO 4): PRINT  INK 2; B
RIGHT 1;AT 20,7+n;"A": NEXT n
 154 IF VAL a$(3 TO 4)<=0 THEN
GO SUB 4250
 156 INPUT "Fleet position";a$(1
)
 157 BEEP .1,15
 158 IF a$(1)<>e$(1) AND ed2=1 T
HEN  LET ed2=0
 159 GO SUB 4500
 160 IF VAL a$(1)=VAL b$(1) THEN
 GO SUB 303
 163 IF a$(1)=f$(1) THEN  GO SUB
 2100
 166 IF a$(1)=c$(1) THEN  GO SUB
 1000
 170 IF a$(1)=e$(1) THEN  GO TO
1900
 172 IF a$(1)<>b$(1) THEN  GO SU
B 4200
 173 LET in=6: GO SUB 9010
 175 IF INKEY$="c" THEN  GO TO 1
85
 180 GO TO 175
 191 LET sc=1
 192 LET in=0: GO SUB 9000
 194 PRINT  PAPER 7; INK 0;AT 10
,2;"PLAYER 2"
 195 GO SUB 1200
 196 IF ed2=1 AND a$(1)=e$(1) TH
EN  GO SUB 4000
 217 PRINT  INK 4;AT 19,14;e$(1)
: FOR n=1 TO VAL e$(3 TO 4): PRI
NT  INK 7;AT 20,6+n;"A": NEXT n
 220 PRINT  INK 7;AT 3,2;"Fleet
2": FOR n=1 TO VAL b$(3 TO 4)
 225 PRINT  INK 7;AT 3,8+n;"A":
NEXT n
 226 IF VAL b$(3 TO 4)<=0 THEN
GO SUB 4250
 227 INPUT "Position of fleet?";
b$(1)
 228 BEEP .1,15
 229 IF b$(1)<>c$(1) AND end=1 T
HEN  LET end=0
 230 LET bb=1: GO SUB 4500: LET
bb=0
 242 IF VAL a$(1)=VAL b$(1) THEN
 GO SUB 303
 244 IF b$(1)=f$(1) THEN  GO SUB
 1650
 247 IF VAL b$(1)=VAL e$(1) THEN
 GO SUB 1800
 255 IF VAL c$(1)=VAL b$(1) THEN
 GO TO 1500
 270 IF VAL b$(1)<>VAL a$(1) THE
N  GO SUB 4200
 290 LET sc=0
 302 GO TO 400
 304 REM combat****
 310 LET shota=VAL a$(3 TO 4)
 320 LET ca=INT shota/2
 330 LET sb=VAL b$(3 TO 4)
 340 LET cb=INT sb/2
 343 LET a=shota-cb
 345 LET d=sb-ca
 350 IF a<0 THEN  LET a=0
 355 IF d<0 THEN  LET d=0
 360 LET a$(3 TO 4)=STR$ a
 370 LET b$(3 TO 4)=STR$ d
 375 GO SUB 4300
 378 PRINT AT 2,8;s$( TO 15)
 380 PRINT  INK 2;AT 2,2;"Fleet
1": FOR n=1 TO VAL a$(3 TO 4): P
RINT  INK 2;AT 2,8+n;"A": NEXT n
 385 PRINT AT 3,9;s$( TO 18)
 390 PRINT  INK 7;AT 3,2;"Fleet
2 ": FOR n=1 TO VAL b$(3 TO 4):
PRINT  INK 7;AT 3,8+n;"A": NEXT
n
 393 GO TO 9200
 400 REM Home planets1&2+n3
 410 LET tn=tn+1
 412 LET d2=d2+1
 415 LET trn=trn+1
 417 LET n1=n1+1
 420 LET b=VAL c$(3 TO 4)
 430 LET b=b+1
 440 LET c$(3 TO 4)=STR$ b
 442 LET b2= VAL e$(3 TO 4)
 444 LET b2=b2+1
 446 LET e$(3 TO 4)=STR$ b2
 448 LET n1= VAL f$(3 TO 4): LET
n1=n1+1
 449 LET f$(3 TO 4)=STR$ n1
```

```
 450 LET in=6: GO SUB 9010
 460 IF INKEY$="c" THEN  GO TO 1
00
 470 GO TO 460
 475 BEEP .3,-2
 500 GO TO 100
 600 REM Choose Home Planet ***
 605 CLS : INK 7
 610 PRINT  INK 4;AT 1,2;"PLANET
S IN THE SAPPRO SYSTEM": PRINT
 620 FOR i=1 TO 10: PRINT AT 6+i
,3-(i=10);i;TAB 6;p$(i): NEXT i:
 PRINT AT 16,18;"Enter 0"
 750 INPUT  BRIGHT 1; PAPER 0; I
NK 2;"PLAYER 1 choose your home
planet";a
 760: LET a$(1)=STR$ a: LET a$(2
)=STR$ a
 770 LET c$(1)=STR$ a: LET c$(2)
=STR$ a
 780 BEEP .1,15
 800 INPUT  PAPER 0; INK 7;"PLAY
ER 2 choose your home planet";a
 810 LET b$(1)=STR$ a: LET b$(2)
=STR$ a
 815 IF a$(1)=b$(1) THEN  GO TO
900
 820 LET e$(1)=STR$ a: LET e$(2)
=STR$ a
 830 BEEP .1,15
 840 RANDOMIZE 0
 850 LET f$(1)=STR$ (RND*10)
 860 IF f$(1)=e$(1) OR f$(1)=c$(
1) THEN  GO TO 850
 870 LET f$(2)=f$(1)
 880 CLS : RETURN
 900 REM players select the same
 home planet
 910 PRINT  INK 6;AT 16,1;"You h
ave selected the same home plane
t as the 1st PLAYER"
 920 PRINT  INK 6;AT 19,1;"Both
PLAYERS must reselect       their
 home planets"
 930 BEEP .1,-6: PAUSE 5: BEEP .
1,-6: PAUSE 5: BEEP .1,-6
 950 GO TO 750
1000 REM Transfer plt1 to flt1
1020 INPUT "Transfer ships? ENTE
R Y/N";d$
1022 BEEP .1,15
1025 IF d$<>"y" THEN  RETURN
1030 INPUT ;"How many?";g
1035 LET tr=tr+VAL a$(3 TO 4)
1037 IF VAL c$(3 TO 4)>g THEN  L
ET tr=tr+g
1040 IF VAL c$(3 TO 4)<=g THEN
LET tr=tr+VAL c$(3 TO 4)
1041 LET cs=VAL c$(3 TO 4)
1048 IF VAL c$(3 TO 4)>g THEN  L
ET c$(3 TO 4)=STR$ (-g+VAL c$(3
TO 4))
1050 IF cs<=g THEN  LET c$(3 TO
4)=STR$ 0
1060 LET a$(3 TO 4)=STR$ tr
1090 LET tn=0
1092 LET cs=0: LET tr=0
1093 PRINT  BRIGHT 1; INK 2;AT 2
,2;"Fleet 1";s$( TO 15): FOR n=1
 TO VAL a$(3 TO 4): PRINT  INK 2
;AT 2,8+n;"B": NEXT n
1094 PRINT AT 20,8;s$( TO 15): F
OR n=1 TO VAL c$(3 TO 4): PRINT
 INK 2;AT 20,7+n;"B": NEXT n
1096 GO TO 9100
1200 REM Clear screen for player
 2
1210 BORDER 7
1220 PRINT  INK 7;AT 9,13;"PLAYE
R 2";AT 10,13;"PRESS";AT 11,13;"
ENTER"
1230 IF INKEY$=CHR$ 13 THEN  GO
TO 1250
1235 BEEP .15,4: PAUSE 1: BEEP .
15,4: PAUSE 1: BEEP .15,8: PAUSE
 1: BEEP .15,8
1240 GO TO 1230
1250 BEEP .07,1: BEEP .07,-1: BE
EP .07,-3
1260 BORDER 0
1270 PRINT  INK 0;AT 9,13;"PLAYE
R 2";AT 10,13;"PRESS";AT 11,13;"
ENTER"
1290 RETURN
1300 REM Clear screen for player
 1
1310 BORDER 2
1320 PRINT  BRIGHT 1; INK 2;AT 9
,13;"PLAYER 1";AT 10,13;"PRESS";
AT 11,13;"ENTER"
1330 IF INKEY$=CHR$ 13 THEN  GO
TO 1350
1335 BEEP .2,9: BEEP .3,5
1340 GO TO 1330
1350 BEEP .07,1: BEEP .07,-1: BE
EP .07,-3
1360 BORDER 0
1370 PRINT  INK 0;AT 9,13;"PLAYE
R 1";AT 10,13;"PRESS";AT 11,13;"
ENTER"
1390 RETURN
1500 REM combat with planet
1505 GO SUB 4400
1530 LET shota=VAL c$(3 TO 4)
```

```
1550 LET ca=INT shota/2
1560 LET sb=VAL b$(3 TO 4)
1562 LET cb=INT sb/2
1568 LET a=shota-cb
1570 LET d=sb-ca
1580 IF a<0 THEN  LET a=0
1585 IF d<0 THEN  LET d=0
1590 LET b$(3 TO 4)=STR$ d: LET
c$(3 TO 4)=STR$ a
1593 PRINT AT 2,2;s$( TO 20);AT
3,2;s$( TO 19)
1600 PRINT  INK 7;AT 2,2;"Fleet
2": FOR n=1 TO VAL b$(3 TO 4): P
RINT  INK 7;AT 2,8+n;"A": NEXT n
1610 PRINT  INK 2;AT 3,2;"Planet
": FOR n=1 TO VAL c$(3 TO 4): P
RINT  INK 2;AT 3,8+n;"A": NEXT n
1611 GO SUB 9200
1613 IF VAL c$(3 TO 4)<=0 THEN
GO SUB 3500
1615 IF VAL c$(3 TO 4)>0 THEN  L
ET end=0
1630 GO TO 271
1650 REM Transfer n plt ,plt2
1670 IF a$(1)=b$(1) THEN  RETURN

1680 PRINT  INK 7;AT 2,2;"Fleet
2": FOR n=1 TO VAL b$(3 TO 4): P
RINT  INK 7;AT 2,8+n;"A": NEXT n
1700 PRINT  INK 4;AT 3,2;"Neutra
l ships": FOR n=1 TO  VAL f$(3 T
O 4): PRINT  INK 4;AT 3,14+n;"A"
: NEXT n
1705 INPUT "Transfer ships? ENTE
R (Y/N)";d$
1707 BEEP .1,15
1710 IF d$<>"y" THEN  GO TO 245
1720 LET tr=tr+VAL b$(3 TO 4)
1721 INPUT "How many ships?";g
1722 IF VAL f$(3 TO 4)>g THEN  L
ET tr=tr+g
1723 IF VAL f$(3 TO 4)<=g THEN
LET tr=tr+VAL f$(3 TO 4)
1724 LET cs=VAL f$(3 TO 4)
1725 IF VAL f$(3 TO 4)>g THEN  L
ET f$(3 TO 4)=STR$ (-g+VAL f$(3
TO 4))
1726 IF cs<=g THEN  LET f$(3 TO
4)=STR$ 0
1740 LET b$(3 TO 4)=STR$ tr
1750 LET cs=0
1760 LET tr=0: LET n1=0
1765 PRINT  INK 7;AT 2,2;"Fleet
2": FOR n=1 TO VAL b$(3 TO 4): P
RINT  INK 7;AT 2,8+n;"A": NEXT n
1768 PRINT AT 3,15;"
   "
1770 FOR n=1 TO VAL f$(3 TO 4):
PRINT  INK 4;AT 3,14+n;"A": NEXT
 n
1773 GO TO 9100
1790 REM Transferplt2****
1800 INPUT "Transfer ships? ENTE
R Y/N";d$
1802 BEEP .1,15
1805 IF d$<>"y" THEN  GO TO 269
1810 LET tr=tr+VAL b$(3 TO 4)
1811 INPUT "How many ships ?";g
1812 IF VAL e$(3 TO 4)<=g THEN
LET tr=tr+VAL e$(3 TO 4)
1813 IF VAL e$(3 TO 4)>g THEN  L
ET tr=tr+g
1815 LET cs=VAL e$(3 TO 4)
1817 IF VAL e$(3 TO 4)>g THEN  L
ET e$(3 TO 4)=STR$ (-g+VAL e$(3
TO 4))
1818 IF cs<=g THEN  LET e$(3 TO
4)=STR$ 0
1840 LET b$(3 TO 4)=STR$ tr
1870 LET cs=0: LET tr=0
1880 LET b2=0
1883 FOR n=1 TO VAL b$(3 TO 4):
PRINT  INK 7;AT 3,8+n;"A": NEXT
n
1884 PRINT AT 20,7;s$( TO 14)
1886 FOR n=1 TO VAL e$(3 TO 4):
PRINT  INK 7;AT 20,6+n;"A": NEXT
 n
1888 GO SUB 1096
1890 RETURN
1900 REM flt 1 v Plt 2
1905 GO SUB 4400
1920 LET shota=VAL a$(3 TO 4)
1930 LET ca=INT shota/2
1940 LET sb=VAL e$(3 TO 4)
1950 LET cb=INT sb/2
1960 LET a=shota-cb
1970 LET d=sb-ca
1980 IF a<0 THEN  LET a=0
1990 IF d<0 THEN  LET d=0
1995 LET a$(3 TO 4)=STR$ a: LET
e$(3 TO 4)=STR$ d
1997 PRINT AT 2,2;s$( TO 20);AT
3,2;s$( TO 20)
2000 PRINT  INK 7;AT 3,2;"Planet
": FOR n=1 TO VAL e$(3 TO 4): PR
INT  INK 7;AT 3,8+n;"A": NEXT n
2004 PRINT  BRIGHT 1; INK 2;AT 2
,2;"Fleet 1": FOR n=1 TO VAL a$(
3 TO 4): PRINT  INK 2;AT 2,8+n;"
A": NEXT n
2006 GO SUB 9200
2010 IF VAL e$(3 TO 4)<=0 THEN
GO SUB 3600
```

```
2020 IF VAL e$(3 TO 4)>Ø THEN  L
ET ed2=Ø
2040 GO TO 173
2100 IF VAL a$(1)=VAL b$(1) THEN
 RETURN
2103 PRINT  INK 4;AT 3,2;"Neutra
l ships": FOR n=1 TO  VAL f$(3 T
O 4): PRINT  INK 4;AT 3,14+n;"B"
: NEXT n
2105 INPUT "Transfer ships? ENTE
R Y/N";d$
2107 BEEP .1,15
2110 IF d$<>"y" THEN  GO TO 167
2115 INPUT "How many?";g
2120 LET tr=tr+VAL a$(3 TO 4)
2124 IF VAL f$(3 TO 4)>g THEN  L
ET tr=tr+g
2126 IF VAL f$(3 TO 4)<=g THEN
LET tr=tr+VAL f$(3 TO 4)
2128 LET cs=VAL f$(3 TO 4)
2130 IF VAL f$(3 TO 4)>g THEN  L
ET f$(3 TO 4)=STR$ (-g+VAL f$(3
TO 4))
2132 IF cs<=g THEN  LET f$(3 TO
4)=STR$ Ø
2140 LET a$(3 TO 4)=STR$ tr
2150 LET tr=Ø: LET n1=Ø
2155 LET cs=Ø
2163 FOR n=1 TO VAL a$(3 TO 4):
PRINT  INK 2;AT 2,8+n;"B": NEXT
n
2165 PRINT AT 3,15;s$( TO 15)
2166 FOR n=1 TO VAL f$(3 TO 4):
PRINT  INK 4;AT 3,14+n;"B": NEXT
 n
2167 GO TO 9100
3000 REM USR Space ships****
3010 FOR n=Ø TO 7
3020 READ j: POKE USR "a"+n,j
3030 NEXT n
3040 DATA BIN 11111000,BIN ØØØ11
100,BIN ØØ111110,BIN ØØØØ1111,BI
N ØØ111110, BIN ØØØ11100,BIN 111
11000,BIN ØØØØØØØØ
3050 RETURN
3499 REM plt lost****
3500 IF VAL b$(3 TO 4)<=Ø THEN
LET end=-1
3510 IF VAL a$(1)=VAL c$(1) AND
VAL a$(3 TO 4)>=1 THEN  LET end=
Ø
3520 IF trn>=Ø THEN  LET end=end
+1: IF end=2 THEN  GO TO 4100
3540 RETURN
3600 REM destruction ofplt 2**
3610 IF VAL a$(3 TO 4)<=Ø THEN
LET ed2=-1
3620 IF VAL b$(1)=VAL e$(1) AND
VAL b$(3 TO 4)>=1 THEN  LET ed2=
Ø
3625 IF trn>=Ø THEN  LET ed2=ed2
+1: IF ed2=2 THEN  GO TO 4100
3640 RETURN
4000 LET o$="  URGENT MESSAGE F
ROM HOME PLANET.... Under attack
..No ships ..HELP!...": GO TO 44
10
4110 FOR n=1 TO 30: LET i=INT (4
Ø-70*RND): BEEP .1,i: LET i=i+SG
N (.5-RND): NEXT n
4120 LET o$="   Congratulations
you have captured the enemies ho
me planet WINNING the game ....T
oday the Planetary system tomorr
ow the Galaxy"
4140 GO SUB 4410
4150 INPUT "Do you want another
game? ENTER Y/N";a$
4160 IF a$<>"n" THEN  GO TO 20
4180 STOP
4210 LET o$="     NO enemy ship a
t this planet": GO TO 4410
4250 LET o$="   Your fleet has b
een TOTALLY DESTROYED never mind
 you can form a new one by retur
ning to your HOME planet": GO TO
 4410
4300 REM Enemy fleet attacking
4305 PRINT  PAPER 7;AT 18,2;s$(
TO 27)
4310 LET o$=" ENEMY FLEET ATTACK
ING    ": GO TO 4410
4400 REM combat enemies home plt
4405 LET o$="   ENEMIES HOME PLA
NET....FLEET UNDER ATTACK"
4410 LET r=1
4412 FOR n=1 TO LEN o$
4415 IF n<=28 THEN  PRINT  PAPER
 7; INK Ø;AT 18,30-n;o$(r TO n)
4420 IF n>28 THEN  PRINT  PAPER
7; INK Ø;AT 18,2;o$(n-27 TO n)
4425 BEEP .07,6: NEXT n
4430 RETURN
4500 REM Draw planet*****
4510 LET pl=VAL a$(1)
4515 IF bb=1 THEN  LET pl=VAL b$
(1)
4527 IF pl=Ø THEN  LET pl=10
4528 PRINT  INK 4;AT 1,4+2*(pl=4
 OR pl=8)-2*(pl=6 OR pl=9);p$(pl
);AT 1,14+(pl=6 OR pl=9);" the "
;pl;("st" AND pl=1)+("nd" AND pl
=2)+("rd" AND pl=3)+("th" AND pl
>3)+" Planet"
```

```
4529 IF pl>7 THEN  LET pl=pl-7
4530 INK pl: GO TO 6600
6100 INK 4: PAPER 0
6110 RESTORE 6200: FOR i=1 TO 15
: READ u,v,w,x: PLOT u,v: DRAW w
,x: NEXT i
6120 FOR n=1 TO 17 STEP 3: RESTO
RE 6210: FOR i=1 TO 2
6130 READ u,v,w,x: PLOT u,v-n: D
RAW w,x: READ u,v,w,x: PLOT u,v+
n: DRAW w,x: NEXT i: NEXT n
6140 FOR i=1 TO 15: READ u,v,w,x
: PLOT u,v: DRAW w,x: NEXT i
6200 DATA 0,0,0,175,0,0,255,0,25
5,0,0,175,255,175,-255,0,15,143,
224,0,14,142,226,0,240,143,0,32,
15,143,0,32,15,32,224,0,15,32,0,
-30,240,6,0,26,14,33,226,0,14,32
,-13,-13,14,33,-13,-13,14,142,-1
3,13
6210 DATA 14,32,-13,-13,14,142,-
13,13,241,32,13,-13,241,142,13,1
3
6220 DATA 14,5,227,0,1,174,254,0
,1,173,254,0,14,169,225,0,14,172
,225,0,241,32,13,-13,241,33,13,-
13,0,2,255,0,14,142,0,27,241,142
,0,27,242,142,13,13,241,33,0,-27
,13,142,-13,13,241,142,13,13,14,
33,0,-27
6410 FOR n=1 TO 90: PLOT  INK 7,
8+236.5*RND,40+95.5*RND: NEXT n
6420 FOR n=1 TO 14: PLOT 240,16-
n: DRAW +n,0: NEXT n
6430 NEXT n
6440 FOR n=1 TO 14: PLOT 241,158
+n: DRAW n,0: PLOT 14,158+n: DRA
W -n,0: PLOT 14,16-n: DRAW -n,0:
 NEXT n
6450 PLOT 0,1: DRAW 255,0
6500 RETURN
6600 LET cz=0: LET cr=31: LET b1
=170: LET a1=170: LET b2=87: LET
 ca2=127: LET n=75:
6625 FOR n=cz TO cr: LET b1=INT
(.5+SQR (cr*cr-n*n)*2)
6630 LET a1=b2+b1/2: PLOT ca2+n,
a1: DRAW cz,-b1
6640 PLOT ca2-n,a1: DRAW cz,-b1
6650 NEXT n
6655 INK 0
6660 RETURN
7000 REM Draw Introduction***
7010 PAPER 0: BORDER 4: CLS
7020 PRINT  INK 4;AT 3,4;"███ ██
█ ███ ███ ███ ███";AT 4,4;"█   █
 █ █ █ █ █ █ █ █ █";AT 5,4;"███
███ ███ ███ █ █ █ █";AT 6,4;"  █
 █ █ █   █   ██  █ █";AT 7,4;"██
█ █ █ █   █   █ █ ███"
7030 PRINT  INK 4;AT 10,4;"AN IN
TER-PLANETARY GAME";AT 12,3;"OF
STRATERGY FOR 2 PLAYERS"
7040 PRINT  INK 4;AT 15,6;" GEOF
F HARRIS 1984"
7050 PRINT  INK 4; FLASH 1;AT 18
,3;"PRESS ANY KEY TO CONTINUE"
7060 FOR n=0 TO 30 STEP 3: PRINT
 AT 0,n;"B";AT 21,1+n;"B": NEXT
n
7070 FOR n=2 TO 21 STEP 3: PRINT
 ;AT n,0;"B";AT n-1,31;"B": NEXT
 n
7075 FOR n=1 TO 30 STEP 3: PRINT
  INK 2;AT 0,n;"B";AT 21,1+n;"B"
: NEXT n
7080 FOR n=2 TO 30 STEP 3: PRINT
  INK 7;AT 0,n;"B";AT 21,1+n;"B"
: NEXT n
7085 FOR n=1 TO 21 STEP 3: PRINT
  INK 2;AT n,0;"B";AT n-1,31;"B"
: NEXT n
7095 FOR n=3 TO 21 STEP 3: PRINT
  INK 7;AT n,0;"B";AT n-1,31;"B"
: NEXT n
7100 FOR n=1 TO 10: BEEP .02,n:
NEXT n
7105 IF INKEY$<>"" THEN  GO TO 7
130
7120 GO TO 7105
7130 FOR n=1 TO 10: BEEP .02,10-
n: NEXT n
7140 BORDER 0: CLS
7150 RETURN
9000 PRINT  PAPER 0;AT 14,12;s$(
 TO 7);AT 7,12;s$( TO 7);AT 13,1
2;s$( TO 10);AT 8,11;s$( TO 10);
AT 9,11;s$( TO 9);AT 10,11;s$( T
O 10);AT 11,11;s$( TO 11);AT 12,
11;s$( TO 11): PRINT AT 18,2; PA
PER 7;s$( TO 28): PRINT AT 1,2;s
$( TO 28): PRINT AT 2,2;s$( TO 2
8);AT 3,2;s$( TO 28);AT 20,7;s$(
 TO 20);AT 19,14;" "
9010 PRINT  INK in;AT 10,22;"PRE
SS C";AT 11,22;"to  ";AT 12,22;"
continue"
9020 RETURN
9100 PRINT AT 2,22; INK 6; FLASH
 1;"TRANSFER";AT 3,22;"COMPLETE"
: RETURN
9200 PRINT AT 2,23; INK 3; FLASH
 1;"COMBAT ";AT 3,23;"RESULTS":
RETURN
```

We, the jury . . .



Juggernaut
CRL
£7.95

It seems that the computer software industry is using its ingenuity and inventiveness to break away from the flight simulator mould at the moment. CRL's program puts you in charge of collecting goods in a large lorry from various points around a town.

If you have ever owned and enjoyed playing with one of those remote or radio controlled cars or lorries then you'll love this program. Unlike the usual simulator the screen has the control and status indicators at the top and, even more unusually, the action screen is not the expected view from the cab, but a bird's eye view of the lorry and the immediate surroundings.

As in all good simulations there are many levels of difficulty, from practice of simple manoeuvres to the full blown exercise of a day's work load.

These options are well graded and the easiest can be mastered fairly quickly. The final level is as complex as any self respecting fan of these games could wish and I wouldn't forsee it being mastered too quickly! My first full game ended in three minutes when I jacknifed the lorry trying to get it into the loading bay.

The instructions are well set out and the on screen options are chosen by the now almost compulsory icon selection system. The graphics are simple but well designed and pleasing, giving information effectively and not adding confusion to the proceedings.

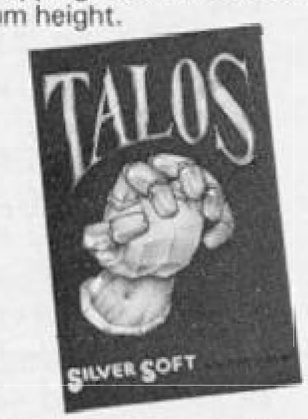
An unusual simulation which is worth having in your games collection.

GRAPHICS ★★★★
ADDICTIVENESS ★★★★★
OVERALL ★★★★

Talos
Silver Soft
£7.95

Though this might seem like a standard maze-chase-zap-and-collect game, the theme and ideas behind this program make it a little unusual. For a start, you are not a whole person, in fact you are not a person at all. The crown of Eternity has been stolen and all the once docile creatures have now become your deadly enemies. TALOS is the only robot left who can retrieve the crown but he has been smashed into seven bits and all that is left that will operate is his hand.

So, you control this lightning throwing hand, avoiding savage snails and grasshoppers and similar attackers as it searches the maze for the rest of its body. To add to your problems is a candle which continually burns down, and burns faster when you are in contact with an insect. This is your 'life-time left' indicator and it can be topped up by passing over various objects. Each time it burns out you lose one of the three lives you began with. Bonus lives can be gained by topping the candle to its maximum height.



This is a fast and furious game which really needs to be played with a joystick for you to stand any chance of success. The screen backgrounds are simple but the sprites are well drawn and animated. I couldn't find my first arm so I only saw my hand movement and this was nicely done. My only criticism is that I wished there was a varying difficulty level as it takes an awful lot of practice to achieve any success, after 45 mins my highest score was 17%.

GRAPHICS ★★★
ADDICTIVENESS ★★★★
OVERALL ★★★★

Red Moon
Level 9 Computing
£6.95

Hello, you say, what's an adventure doing in the arcade reviews? Simple, I reply, it arrived too late for Brian's section and was too good to leave until next issue.

I HATE Level 9. Why? The reason is simple, every time they bring out a new adventure I get so absorbed in it that I neglect my work, eventually I have to leave it and then never get round to completing it. I have not managed to succeed in any of their games yet.

Level 9 have so far concentrated on unusual scenarios for their adventures and have gained a reputation for not only being good but also for being different. Red Moon is their first Magical/Traditional theme and as such it sets a new standard for this type of program. It is hard to define what makes it so good, could it be the quickly drawn pictures for each of the 200+ locations? or the atmosphere created by the literate text? or perhaps the well designed and created storyline?

I certainly find the speed with which you can get into the game before encountering those mind-twisting problems irresistable. And, when you do die there is the possibility that I may be reincarnated with all the objects I've found so far — although this doesn't always work.

Excuse me, I just had an idea about how I may be able to get rid of that Giant Rat . . .

GRAPHICS ★★★★★
ADDICTIVENESS ★★★★★
OVERALL ★★★★★



Pole Position
US Gold / Datasoft
£7.95

This is an accurate computer version of the arcade hit of the same name, the graphics have been faithfully reproduced and, as far as is possible, the sounds are the same.

Pole Position is a race driving game. I say game rather than simulation as the only controls are left and right plus forward and back for Hi/Low gear change and brakes.

The first section has you steering your car, which you see as a rear view at the bottom of the screen, left and right as the track scrolls towards you, there are other cars which appear and have to be avoided as you overtake them.

Provided that you get a lap speed of 73mph or better, you then go on to the actual race itself and do the same again in the main race.

Hitting another car or one of the billboards causes your car to explode. If this happens you get a new car and start from this position, the only real penalty is the time you lose.

It is hard to say why, but I wasn't too excited by this program. All the arcade features are there, but going round and round the track didn't seem too challenging — OK, I had my share of crashes, but I couldn't maintain a lasting interest.

If you are a fan of the arcade version then this will be a must for you, I suspect most Spectrum gamers will find there is not enough variation in the gameplay to keep them engrossed for long.

GRAPHICS ★★★
ADDICTIVENESS ★★
OVERALL ★★★



DATABASE SOFTWARE

Red Arrows
Database Software
£8.95

OK Biggles, chocks away! And with that cry I flew straight into the ground.

There are a lot of flight simulators about already, so what are Database offering to entice you to buy their program? Well, for a start, instead of just flying around aimlessly you have the excitement of flying in formation with the Red Arrows (hence the name), but first you have to attain the high degree of skill required. You can jump straight in and try, but I almost guarantee you'll fail.

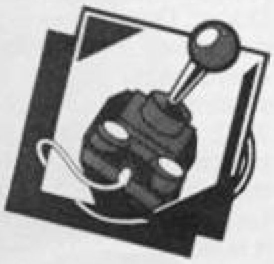
The options are many, not the least important are the training sections with four degrees of difficulty or the 'Help' messages given to novices. The flight manual gives a great deal of information both about the Red Arrows and the Hawk T, Mk 1. Hints and tips are provided, and indeed the whole program goes out of its way to help you get to grips with the business of flying competently.

Once you have mastered the basic techniques you can then practice each manoeuvre or go for the show, all with options to make the difficulty level easier or harder.

This program has all the usual tricky combinations of keys to be mastered, and joystick if required, and is every bit as complex as its competitors. Where it scores is in the relatively gentle introductory stages and the final aim of a performance of aerobatics. The graphics are very good but suffer from the same jerkiness as other programs of this type. Highly recommended.

GRAPHICS	★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★★











Nick Faldo Plays The Open
Mind Games
£9.99

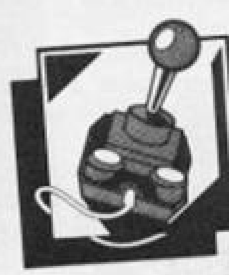
Golfing simulations are not new, many very respectable versions have been published in *ZXC* and other mags. So what is this program going to offer to warrant £9.95 I asked myself, apart from NF's endorsement? Three things actually, first a very accurate representation of The Royal St. George Golf Club in Sandwich where the 1985 Open Championship was held. The handbook also contains a hole by hole description and suggestions of how to attempt them.

Second is the input system which is performed completely by an Icon system, this allows joystick operation and very precise direction/power selection.

And finally, the birds' eye scrolling map of the course is most impressive, not to mention the additional control when the little golfer swings at the ball which can add extra power and/or direction (including slicing) to the stroke, giving it a realistic feel.

Then there are the comments and 'advice' of the caddy on your performance or choice of club.

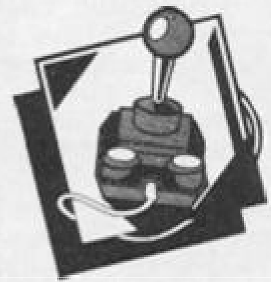
OK, so there are lots of nifty features but does it play well and is it fun? The answer is yes, the split screen icon system works well and you can improve with practice. The first time around, my score was 121, second time 86 — still plenty of time for improvement.





The manual is nicely written and produced, but as with every program of this nature there is no explanation of the merits of different clubs. For the complete ignoramous such as I, when do you use a Wood? When an Iron?

GRAPHICS	★★★★
ADDICTIVENESS	★★★★★
OVERALL	★★★★



Dambusters
US Gold/Datasoft
£9.95

This is a complex mix of strategy, simulation and arcade games and, surprisingly, it works. It is a very complicated game and will challenge even the most experienced player.

You are in charge of a RAF Lancaster Bomber of 617 Squadron and your mission is to destroy one of three dams, the Moehne, the Eder or the Scorpe. From the main menu you have a large selection of options; three skill levels, readout mode (digital or analogue) and the starting mode; Practice, In Flight, Take Off.

Each member of the flight crew is you! You switch from one display to another performing the tasks of Pilot, Tail Gunner, Bomb Aimer, Navigator, First Engineer, Second Engineer and finally checking on your status, damage and score.

I ran the game several times to get a look at the screens and found them interesting but in the main rather confusing. This was due to the fact that the preproduction version of the program supplied to us came without the "comprehensive flight instructions, maps and documents" that are sold with the program. (*We did receive a full production version of the game, but it arrived too late to alter the review — Ed*).

With a game of this complexity all I could do was get an overall impression, and that impression was very favourable.

GRAPHICS	★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★

Metabolis
Gremlin Graphics
£6.95

The Kremins have converted all human life into the lesser forms of mice and birds. You, being different, ended up transformed into a bird but retained your human brain and so, once again, it's up to you to save the human race.

To survive and complete your mission you have to find the serum which will change you back into human form, destroy the area with the four bits of nuclear fuel which you find lying about, and make your escape. Some of the life forms will help, though most will attack you. Also you have to find and eat food regularly and watch your heart as the transformation left it weak.

Essentially this is a maze dodge-chase-and-collect game, but it is well written and all the screens that I managed to visit out of the 150 in the game, were colourful and well designed. It is a fast and furious game, and some of the graphics reminded me of the White Lightning program, but a wide variety of techniques have been used. Your bird identity is lovely, it is such a cute and lovable graphic that it seemed a pity to change it back into a mere mortal.

Although not state of the art it is a challenging and respectable program which deserves consideration for a place in your games collection.

GRAPHICS	★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★





Hyper Sports
Imagine
£7.95

Look out joystick, here we go again! In this variation of the track and field type of program, a new set of events are presented to test your wrist stamina and joystick ruggedness.

The sports presented in this section are:

1. Swimming, a rapid L/R movement plus pressing fire at the correct time when prompted to breathe.
2. Skeet shooting, no aiming required just L/R when the skeet is in sight. Good reactions required.
3. Long Horse, this requires judgement on when to press the fire button followed by rapid L/R action.
4. Archery, judging when to press fire and how long to hold it down.
5. Triple Jump, rapid L/R movement followed by holding button down for the best amount of time.
6. Weight Lifting, L/R, timing of button press then more L/R action.

I was never a fan of the original variations of Track and Field but to be honest I really enjoyed the game and probably spent longer on it than on any other game I've seen recently.

The graphics are of the high quality that we expect from both this type of program and Imagine, and there are some nice touches. I enjoyed watching the man bounce along on his head when I misjudged his Long Horse jump!

Great for fans of sports games, it should also make a few more converts. My only criticism is that I couldn't practice each event separately.

Joystick manufacturers will love it.

GRAPHICS	★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★



Frank Bruno's Boxing
Elite Systems Ltd
£6.95

Sports simulations seem to be the in thing at the moment, and I have learnt more sporting personalities' names via these programs than I've ever known.

There are one or two similar programs out at the moment, and it is up to the individual to find the one which suits them, however this one has a satisfying variation of control options and eight individual opponents, each with their own fight techniques.

A joystick option is provided but you still need to use some keys, in fact I found it to be one of the few games where the keyboard was preferable. Operating keys are not redefinable but have been chosen well: 1-Guard up, Q-Duck, A-Guard down, U-Dodge left, I-Left punch, O-Right punch, P-Dodge right and bottom row keys-Right hook/uppercut.

The eight boxers represent different countries and their names are quite amusingly reflective of the nationality, good job the race relation board didn't notice!

The easiest boxer, Canadian Crusher, loads with the program but to meet other opponents you have to defeat him to get a code which will allow you to load in the next fighter from side B of the tape. Good fun, but perhaps not the variety or sophistication of action than many gamers are used to, the graphics are impressive, with a view over the back of FB who you control, and the boxers are well animated.

GRAPHICS	★★★★
ADDICTIVENESS	★★★
OVERALL	★★★





Southern Belle
Hewson Consultants
£7.95

I could never really cope with flight simulations, when I were a lad I wanted to be a train driver anyway! So now my youthful dreams can be realised, on the computer screen at least, thanks to this program. In a fit of inspirational genius, Hewson decided to produce this simulation of driving a King Arthur class 4-6-0 steam train from London to Brighton.

Now, in itself this may not sound particularly arduous but when you realise all the variables that need to be balanced, Water Steam, Whistle, Regulator, Cut Off, Vacuum Brake, Blower, Injectors, Dampers, Firehole Doors, Gradients, Signals and a schedule, it's not that simple. Fortunately there is a wide selection of control levels — 132 in all, plus some unexpected problems which are thrown in at the hardest level, just when you thought you'd got it sussed!

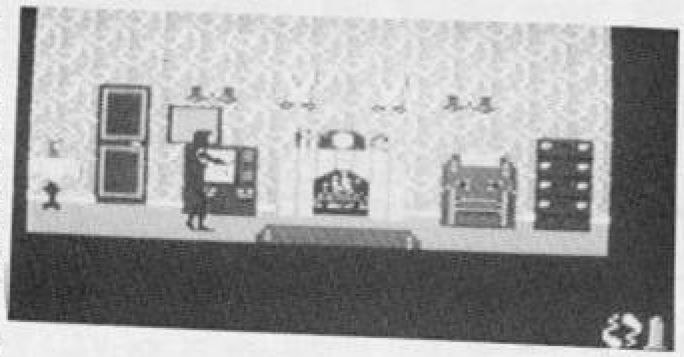
I was, at first, a little disappointed by the graphics, which present plus line drawings of the main landmarks in 3D perspective, but as I played the game I realised they were perfect. Not too much action to detract from the main job of operating the engine, but just enough to give you visual information of your position. The animation is effective, but is a little jerky.

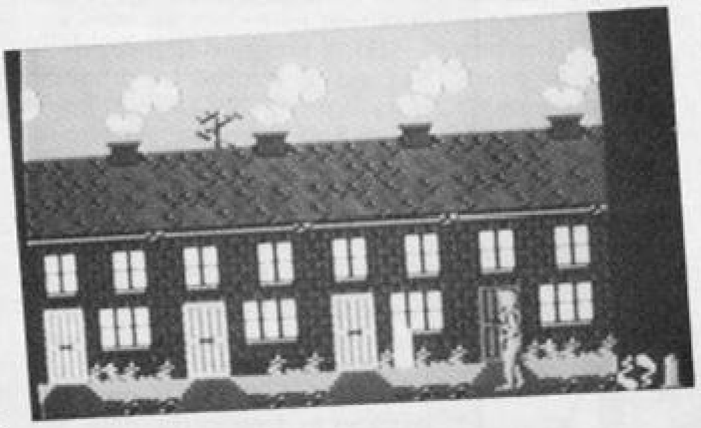
Not only will old hands find a challenge, but those who do not have the experience of this kind of simulator program should consider this one for the ease with which you can gain experience, even the manual is written in a sensible, easy to read way.

A well balanced, challenging and beautifully conceived program.

GRAPHICS	★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★★











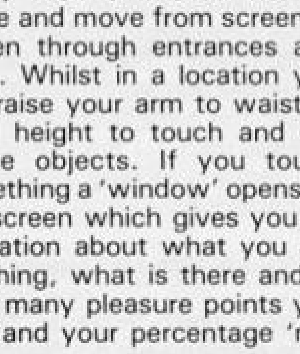


Frankie Goes to Hollywood
Ocean
£9.95

It's been a long time arriving but was it worth the wait?

This program is unusual in its creation, not just a simple matter of licencing the name but an actual joint venture between all interested parties. The end result shows what this kind of cooperation can produce — it's brilliant!

The concept bears a vague resemblance to *Deus Ex Machina* but the implementation is totally unique. You control, via the keyboard or joystick, a large, well animated human figure and move from screen to screen through entrances and exits. Whilst in a location you can raise your arm to waist or head height to touch and examine objects. If you touch something a 'window' opens on the screen which gives you information about what you are touching, what is there and/or how many pleasure points you gain and your percentage 'real person' score. The various parts that make this up are shown graphically at the bottom right of the screen.



When the window is open, and if there are objects there, a movable hand points to them and by pressing 'fire' they can be taken. To use carried objects you have access to your inventory window at any time, and can use the hand to select the object and use it by pressing 'fire'. Icon graphics are used very successfully for this section.

Built into the program are several arcade style sections, and there are over 60 problems to solve, including a whodunnit murder mystery, raid over Merseyside and shooting gallery sequence. The packaging is excellent and the game itself is easy to get started and is probably the most addictive that I've ever played.

We have decided that we are going to extend our rating system with a special six star award for programs which we consider to be state of the art in one way or another. Frankie gets it! (and so should you).

GRAPHICS	★★★★★
ADDICTIVENESS	★★★★★
OVERALL	★★★★★★

Popeye
Dk'Tronics
£7.95

First announced just under a year ago, but delayed due to the chronic Spinach shortage, *Popeye* finally makes it to the computer screen.

The first thing that strikes you about this game is the graphics, they are HUGE, each character is half the height of the screen, lovingly detailed and animated, and sets a new standard for character graphics. The insert tells you precious little about what to do and, as I would have been at a loss without the reviewer's sheet, I will pass on a few tips to you — ignore the next bit if you don't want to know.

You have to collect 25 hearts and take them to Olive, these have to be taken to her in stages as you have to keep her interest, indicated by the lovemeter. Collect Spinach for recovery when knocked over, keys to open their particular doors, and other objects for future use. Eight objects may be carried at one time so be selective.

The most difficult part is the imaginary 3D or 'layer' system of graphics. The graphics are in fact 2D but there are three or four backward and forward planes which you can be on. For instance, if Popeye is facing you and you walk him forward he will seemingly walk on the spot. For the purpose of the game, he has effectively walked towards you. The reverse also applies.

All moving objects are dangerous if they collide with you, this happens if you are in the same plane, otherwise they will pass behind or in front of you. As well as Up/Down meaning forward and back it can also mean special things in certain circumstances such as climb up or down a rope.

There can be no such thing as a perfect program and the biggest fault I can find with this one is the slowness of the action. However there is a direct relationship between the size of an object and the speed at which it can be animated. All in all I like the size and detail of these graphics too much to be very critical of the speed.

Popeye is a program which pushes the limitations of the Spectrum back yet one more step.

GRAPHICS	★★★★★
ADDICTIVENESS	★★★★
OVERALL	★★★★★

Microdrive Adventures

By Alan Giles



Spectrum microdrives offer at least 85K bytes of storage space, which a lot of programmers, myself included, would like to be able to use, to expand upon the 48K bytes of RAM in the Spectrum itself. Whether you are writing adventures, or more serious business software, your requirements from the microdrive will probably be very similar, you need to be able to read any 512 byte microdrive sector at random, possibly modify it and write it back in the same place on the tape. At first I thought I would need a lot of machine code to do this sort of thing, but I discovered that an amazing number of things can be done from BASIC with a few PEEKs and POKEs into the microdrive channel (described in Appendix 3 of the manual that comes with the microdrive) — this also has the advantage of avoiding any problems associated with the different versions of the Interface 1 ROM. I also found what I think is the first serious use of the VAL$ function.

So, a program first needs to set up a microdrive channel area. From BASIC the command needed is something like:

```
OPEN #4;"m";1;CHR$ 0 + CHR$ 255
```

I use the rather obscure file name CHR$ 0 + CHR$ 255 as it is unlikely to exist, and, to create a complete 'map' with all the free sectors on the cartridge marked as free in the map, we need to open a file for writing, as the Interface 1 then checks every sector on the cartridge. If you only intend reading the cartridge then a correct map is not essential and you can open a file for reading in order to create the required channel area (the map then only records the free sectors found before the desired file was found).

Now, we need to find where this channel is in memory. As the opening and closing of other channels may well move this channel around in memory, this figure is something that may need calculating quite often, so I have defined a function to do it:

```
DEF FN c(n) = (PEEK
(n* 2 +23575) + PEEK
23632)* 256 + PEEK
(n* 2 +23574) + PEEK
23631 - 1
```

This function takes the start of the channels area, stored at 23631, and adds the offset for stream 'n' from the streams table, minus 1 (offset zero is used for closed streams, so 1 is added to the offsets as stored in the table).

```
LET c4 = FN c(4)
```

will now set up a pointer to the channel area we OPENed, which we can safely use until another channel is opened or closed when we may have to re-execute the above line. Using 'c4' is faster than using 'FN c(4)' because of the time taken to compute a function.

We can now look at some of the channel details of the cartridge in the microdrive:

```
PRINT "Number of sector just read:";PEEK (c4 +41)

PRINT "Cartridge name:";:FOR i=c4 +44 TO c4 +53:PRINT CHR$ PEEK i;:NEXT i
```

HOW MICRODRIVE COMMANDS AFFECT THE MICRODRIVE MAP

The start of any microdrive command sets all sectors as 'in use', so it is up to each command to discover its own free sectors that it needs. MOVE, ERASE, FORMAT, LOAD, VERIFY and OPEN for reading can leave the map in a state where not all free sectors are recorded in the map, so, subsequent writes may report the error 'Microdrive full' before all the actually free sectors are used up. CAT is a useful command to generate a fresh map of free sectors, as long as there are less than 50 unprotected files on the cartridge.

OPEN	for writing/SAVE: all free sectors are flagged as such.
OPEN	for reading/LOAD/VERIFY: Only those sectors found before finding the required file are flagged as free.
MOVE:	If the source has to be OPENed, free sectors found before the file are flagged as free, then if the destination file has to be OPENed, all free sectors are flagged as free.
ERASE:	Only those free sectors found before all the file to be ERASEd, plus those of the file ERASEd, are marked as free.
CAT:	Only those free sectors found before the fiftieth unprotected file name is found are flagged as free.
FORMAT:	All sectrors are flagged as in use.

Table 1: Microdrive Channel Contents Ix = Do not POKE, N = POKE has no lasting effect

Number Of Bytes	Position in Channel	Name	Description Or Use
X2	0	8	Directs PRINT through Interface 1 ROM
X2	2	8	Directs INPUT/INKEY$ through Interface 1 ROM
X1	4	'M'	Confirms this is a microdrive channel
X2	5	MCHAN-OUT	Address of PRINT routine in Interface 1 ROM
X2	7	M-INPUT	Address of INPUT/INKEY$ routine in IF1 ROM
X2	9	595	Length of channel data, so the start of the next channel can be found
2	11	CHBYTE	Counter of how far through the current record you've got PRINTing or INPUTting. Can be useful to reset it to zero (so you can start again), 511 (so the next PRINT will trigger a sector write), 512 (so the next INPUT will trigger a read).
1	13	CHREC	Current file record number. Useful numbers: zero (to fit in with our random filing system), 255 (so the next INPUT reads record zero).
10	14	CHNAME	Ten bytes of filename filled out with spaces. Checked against the sector read when reading, or copied into a sector to be written before it is actually written.
1	24	CHFLAG	254 if the file is open for reading, 255 for writing. This must be set to the correct value for the current operation, or you get an error.
1	25	CHDRIV	Microdrive number - only really useful if you've got more than one.
2	26	CHMAP	Pointer to map for this drive.
N12	28	preamble	Only used by FORMAT, so free for your use.
N1	40	HDFLAG	1 confirms that a sector header was read.
N1	41	HDNUMB	Sector number of last sector read/written.
N2	42	HD-ID	Two byte pseudo-random number which may help you to tell the difference between two cartridges which have the same name. However, the number is not particularly random, so it is not a guaranteed way of checking for different cartridges.
N10	44	HDNAME	Cartridge name of the last cartridge read/written.
N1	54	HDCHK	Checksum.
X12	55	preamble	Written to the cartridge every time a sector is written. If it does not consist of 10 bytes of zero and two bytes 255, you will not be able to read the sector back.
1	67	RECFLG	The sum of: 2 if last record in file, 4 if a SAVE/LOAD file type.
N1	68	RECNUM	CHREC is copied here before a record is written.
N2	69	RECLEN	CHBYTE is copied here before a record is written. Otherwise records total number of valid bytes in the sector.
N10	71	RECNAM	CHNAME is copied here before a record is written.
N1	81	DESCHK	Checksum.
512	82	CHDATA	The 512 bytes of data created by PRINT or read by INPUT/INKEY$. If this is record zero of a SAVE/LOAD file, the first 9 bytes have special meanings.
N1	594	DCHK	Checksum.

PRINT "Sector is from file:"; FOR i=c4+71 TO c4+80: PRINT CHR$ PEEK i;:NEXT i

PRINT "Record number in file:"; PEEK (c4+68)

PRINT "Open for ";"reading" AND PEEK (c4+24)=254; "writing" AND PEEK (c4+24) =255;

You may wish to check that the file has been opened for writing, so that you know that the map is correct.

Creating a Data Cartridge

The best way of using a cartridge to allow random access to any sector is to construct files which only occupy one sector — which limits them to 512 bytes. This means that you may need to create more than 50 files on a cartridge — perhaps as many as 170 or more. This number of files will confuse CAT unless many of them have names beginning with CHR$ 0, when CAT will ignore them. So, I tend to use file name CHR$ 0 + CHR$ n where 'n' is the adventure game location number, or just a conceptual record number in a business program superfile. This also means that changing the file name only involves one POKE rather than ten, in the style:

POKE c4 +15,n

Note that you change the name in the channel variable CHNAME, and not in RECNAM, because as the Interface 1 ROM prepares to write a sector it copies CHNAME to RECNAM.

To allow these files to be deleted quickly, it is probably best to make all the records 'end of file' records. This could be done in BASIC using the CLOSE command, but this destroys the channel area, which would involve us in another OPEN if we want a channel area again, which will take at least seven seconds. So, rather than doing this we can POKE the end of file flag with:

POKE c4 +67,2

To cause a record to be written to microdrive we then have to write exactly 512 bytes to it rather than using CLOSE, this may involve filling out the record with bytes which do not cause problems. If PEEK (c4+13) is still zero when the proper data has been PRINTed to channel 4, the number of filler bytes needed is 512-PEEK (c4+12)✱256-PEEK (c4+11). If PEEK (c4+13) is not zero then you have already tried to PRINT 512 or more bytes to the file, indicating a problem unless PEEK (c4+13)=1 and both PEEK (c4+11) and PEEK (c4+12) are zero, when exactly 512 bytes have been written.

You will notice that I have been using PRINT type files rather than SAVE/LOAD type CODE or DATA files. I suppose I could have used such files, but if I did I would waste the first nine bytes of the file, having to record the equivalent of information from system variables HD-00 to HD-11. If you want to do this, the only other change needed is to:

POKE c4 +67,6

rather than 2 as above.

There are two approaches to PRINT type files. The simplest to use with an adventure is the one which expects to use MOVE to transfer the text of a location description from microdrive to the screen in the style:

MOVE "m";1;CHR$ 0 + CHR$ n TO #2

This approach allows you to use all the obscure Spectrum screen control characters, changing colour, ATs, TABs, you name it — which means you have to be really careful with the filler bytes, I use CHR$ 23 which has the disadvantage of being the TAB code, typically moving the print position out to column 23 but hopefully doing nothing else to upset the display. CHR$ 9 is another interesting "do nothing" as in most Spectrums it doesn't PRINT anything at all — unfortunately in the Scandinavian version of the Spectrum ROM it moves the print position one place to the right, and if you have a few hundred of these, things could look a bit odd. *(OK, how many of you didn't know that there's a Scandinavian Spectrum ROM floating about? Remember, you read it first in ZXC — Ed.)*

The other approach to PRINT type files is to expect to read them using INPUT. This means you cannot include CHR$ 13 in any item, as CHR$ 13 is used to terminate INPUT. Even TABs to column 13 and ATs involving either row or column 13 will cause problems. You are also

Table 2: Save/Load file initial 9 bytes

Number Of Bytes	Position in File	Description Of Contents
1	0	File type: 0=program, 1=numeric array, 2=string array, 3=code file.
2	1	Data block length.
2	3	Data block start address.
2	5	Program length without variables, or array name (80-9F hexadecimal if numeric, C0-DF hex if a string array).
2	7	Autostart line number, or FFFF hex if not autostarting.

▶

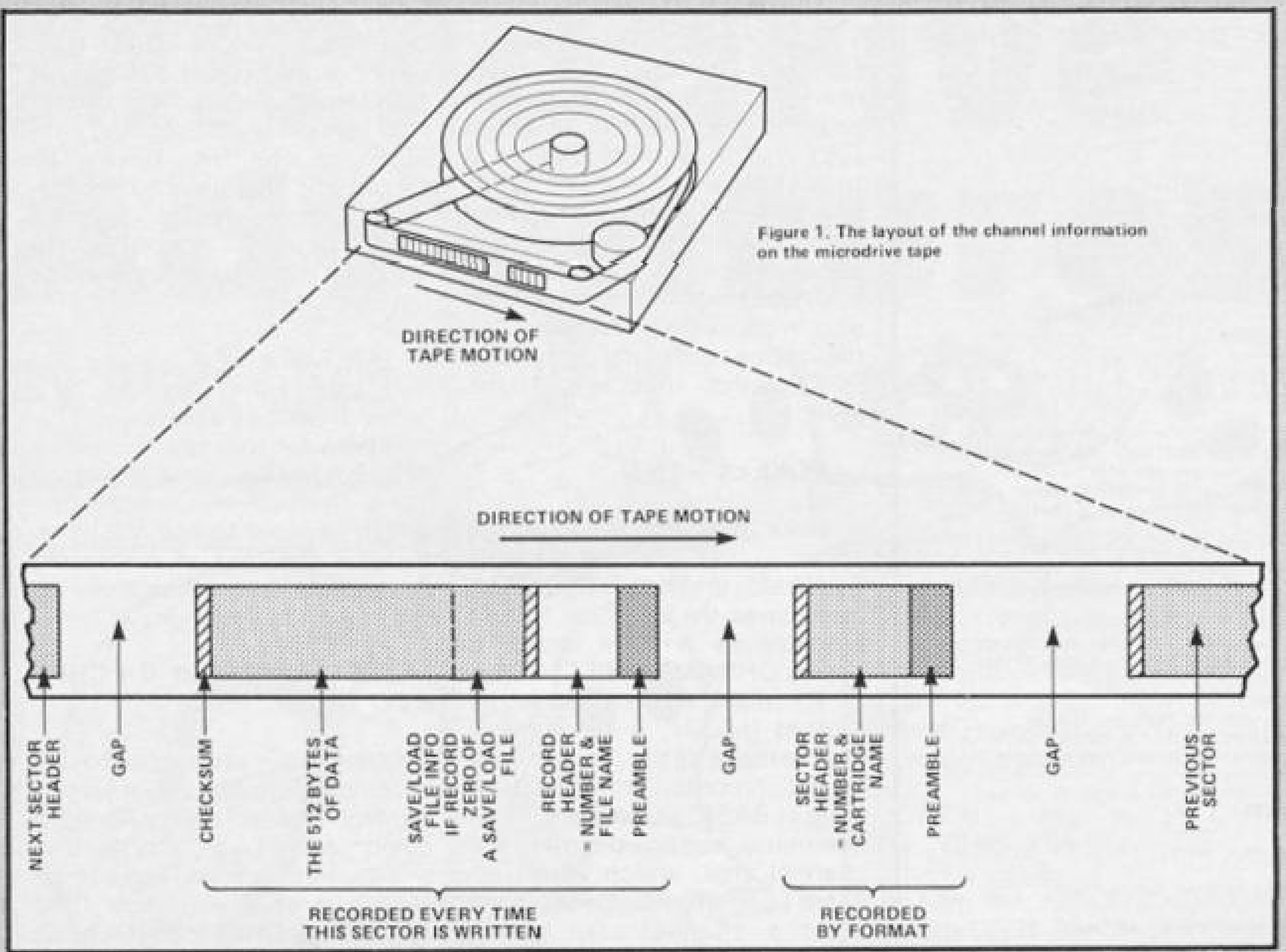


Figure 1. The layout of the channel information on the microdrive tape

advised to use INPUT LINE to avoid problems with the quotation mark sign. With this approach you will never read the filler bytes so the value does not matter, indeed you can fill using the commands

```
POKE c4 +11,255:POKE
c4 +12,1:PRINT #4.
```

It is in this use of the file that you can read strings which you might like to process with the VAL$ function, but more of that later.

Sectors

In the meantime let's clear up a few details of the process of writing sectors. When a sector has been written, PEEK (c4 +41) tells you exactly where on the cartridge the data has been written — remember that these sector numbers count down from some high number to one, then you start going around the tape again. The amount of tape travel involved while the microdrive motor builds up to speed and then slows down again means that successive sectors will probably be recorded 7 or 8 sectors apart, assuming that such a sector was listed in the map as free for use. To prepare to record the next record of data you need to POKE c4 +13,0 because we want to do record zero of the next file, PEEK (c4 +11) and PEEK (c4 +12) should now be zero because the last record was filled out to 512 bytes, and the only other thing you have to do is to POKE c4 +15 with the next file number.

So, slowly but surely you can build up a collection of sectors containing adventure location descriptions, business file records or whatever, by PRINTing #4 the details interspersed with a few POKEs to get the file names right and to trigger the writing of data. If you happen to make a mistake you can use ERASE 'm';1;CHR$ 0 + CHR$ n to delete the item and start again, providing you POKE c4 +15 with something other than 'n' ('n+1' for example) to avoid a ''Reading a 'write' file'' error. Or you can manipulate the map to cause the sector to be overwritten next time.

Manipulating The Map

Manipulating the map is quite devious, but business programs which allow data to be altered as they run will need to use this technique to force the new version of a sector to overwrite the old. First we need to find the address in memory of the map, maybe with another function:

```
DEF FN m(c) = PEEK
(c+27)* 256 + PEEK (c+26)
```

where 'c' us a channel address found by FN c(n).

A map is 32 bytes long, and the first thing you might find useful to do is to save the current map contents in case you need it later:

```
LET m4 = FN m(c4)
DIM m$ (32)
FOR i=1 TO 32:LET
m$ (i) = CHR$ PEEK
(m4 +i-1):NEXT i
```

To force the next record of data to be written to a given sector the map entry for that sector needs to be the only one set to zero. If the required sector number is 's' then:

```
FOR i=0 TO 31:POKE
m4 +i,255:NEXT i
POKE m4 + INT
(s/8), 255 -2 ↑ (s-8_
* INT(s/8))
```

To check if sector 's' is free for use in the map in 'm$':

```
LET temp = CODE
m$ (1 + INT(s/8))
LET bit=2 ↑ (s-8 * INT(s/8))
LET use = INT (temp/bit)
-2 * INT (temp/2 /bit)
```

'use' will be zero if the sector is free for use, or one if it is already in use or otherwise unavailable (for example, near the microdrive tape splice).

Having found a free sector and decided to use it, you ought to record in 'm$' that you have done so with:

```
LET m$ (1 + INT(s/8)) = CHR$
(CODE m$ (1 + INT(s/8)) + bit)
```

Map manipulation looks complicated, and you can end up overwriting the wrong sector if you are not careful, but remember that you only need to do this if you have for some reason to force a particular file into a particular microdrive sector, or if you want to reduce the two or more tape revolutions involved in an ERASE and rewrite to the single revolution of an overwrite.

Multiple copies of Files

In an adventure you might decide that to reduce the seven second maximum time taken to find and read a sector, you will record a location description twice or even more times on the same tape. This will only help access speed if the copies of the file are well spread around the tape rather than being collected together, so on an 85K tape to record two copies half the tape length away from each other you will be looking for sector numbers 85 apart. to record the same data as last time in another sector, set up the map as necessary POKE c4 +13,0: POKE c4 +11,255: POKE c4 +12,1 and PRINT #4; PEEK (c4 +593); — that's the final character of the record, be it a filler byte or whatever.

Another way to spread things round the tape without manipulating the map, is to use the fact that you can PEEK the sector number of the last operation, and so after doing 11 or 12 files tell that the tape has moved about half way round, and then record the same 11 or 12 files again.

An interesting use of such multiple file copies is to actually record different information in different copies of the same file. Then, when the file is read, the information you read is selected at random from the different versions of the file depending on the tape position. This sort of thing can make an adventure game really interesting if it's handled correctly. Of course, the more you use these multiple copies the more tape gets used,

and to give a game a large number of locations the present system I'm describing needs a tape sector for every location, so you probably only want to repeat the frequently accessed locations.

A use for VAL$

Alternatively, you will probably find that a number of adventure locations can be described in much less than 512 bytes, allowing two or more descriptions to be fitted in some sectors. This approach is obviously not suited to description display using MOVE, but can be used with INPUT. Either you can decide to PRINT extra lines in the file which are only INPUT when your adventure program knows that they are there (I leave the details to you), or you could use VAL$.

VAL$ is a Spectrum function which was added to the ROM as much because it was easy to add as for any useful function it can perform. It takes a string and evaluates its contents as an expression, just like VAL, the difference is that it gives an error if the answer is numeric, only working if the result is also a string. This can lead to the original string needing a large number of repeated quotation marks to ensure that the final result is still a string. You might imagine writing to a microdrive file something along the lines of:

```
("Description 1" AND d=1)+
("Description 2" AND d=2)
```

You read this back with INPUT #4;LINE a$ and display on screen with PRINT VAL$ a$. As long as 'd' is set to the correct value, the appropriate description will be pulled out of 'a$'.

Another advantage of using VAL$ to process strings before printing, is that you can restore screen movement control codes, using CHR$ 13 and such like to represent them, and you can produce descriptions longer than 512 bytes by referring to other strings loaded with the program, for example:

```
z$ +"more description details"
```

where 'z$' is description preamble of some sort, built into your adventure program.

Reading data from the cartridge

The adventure command 'LOOK' is often used to redisplay a location description. It would be silly to have to reread the description from microdrive in order to display it, when it is still in memory from the last time it was read. It is a simple matter of:

```
POKE c4 +11,0:POKE
c4 +12,0
```

to reset the pointer so that you can INPUT the same description again without triggering the microdrive motor into action.

To take this idea further, to reduce the number of microdrive accesses made by your program you could OPEN several streams to different location files as you go along. Every OPEN stream creates its own 595 byte microdrive channel area, which includes the 512 bytes into which the location details are loaded. These 512 bytes are not lost from memory until the stream is CLOSEd or CLEARed, so it is possible to reset the channel pointer CHBYTE as above, perhaps modified as here:

```
POKE FN c(n)+11,0:POKE FN
c(n)+12,0
```

So, if your program holds details of which stream is keeping a copy of which location description, it can first check whether a description is in memory before going to the time consuming task of re-reading it from microdrive.

When reading from files the continual CLOSing and OPENing we were worried about during the file writing does not take much more time than that which can be achieved using POKEs, the only delay is the time taken copying all your program backwards and forwards in memory, which can be bypassed with the following POKEs:

```
POKE     c4 +11,0:POKE
c4 +12,2:POKE
c4 +13,255:POKE c4 +15,n
```

This sets up the channel as if all 512 bytes of record −1 of file CHR$ 0 + CHR$ n had already been read, so that when you use INPUT or INKEY$ the microdrive motor is triggered and record zero of the required file is read into memory. (This same approach can be used to read SAVE/LOAD files from BASIC, as it bypasses the file type checking which only happens during OPEN — so you OPEN any other file then change the file name and make sure you are in READ mode with POKE c4 +24,254 before setting the pointers as above.)

Conclusion

Although I have only given a few BASIC program lines in this article, I hope that I have managed to convey a feeling for the variety of devious microdrive activities which can be undertaken by BASIC programs, and I hope I have encouraged you to go out and write a world-beating adventure. . .

Alan Giles



ZX81 Chatterbox

ZXC introduces a new, regular column from that saviour of the '81, Software Farm's Julian Chappel

'Ah!', I hear you all say. 'a new section in my favourite mag. What's it all about?'

To give you a subtle hint, if you are not the proud owner of that wonderful little machine the ZX81 – then bog off! We don't want any of your attributes problems here! This column is solely for the benefit of that much neglected group of people with lumpy graphics, flat keyboard and silent music – and are proud of it!

Right lads and lasses! Now that we've got rid of the opposition, let's get down to business! If you own a ZX81 and haven't been living at the bottom of a hole in the middle of the Sahara Desert wearing ear muffs and blinkers, you will have heard of the Software Farm. Having heard of the Software Farm the chances are that you will have come across the S.F.S.C. (Software Farm Software Club) which, run by an elusive character known as the Cosmic Cockerel, caters exclusively for owners of this little block wedge. It has brought a ray of sunshine into the otherwise lonely existence of ZX81 owners everywhere, by allowing them to air their views, comments, gripes and grumbles; swap ideas; answer each other's problems and generally make contact with other ZX81/TS1000 owners around the world. No longer do you have to sit all alone in a dark corner of the local computer club, sobbing gently and amusing yourself by throwing bricks at Spectrum owners. Instead, the satisfaction of finding a whole group of people with the same interest fills you with the warm feeling of being wanted. A welcome break from the battle of the high street where the question "Do you stock any ZX81 titles?" usually brings a response that would make anyone feel like a bright purple bug-eyed monster from Mars.

This unhappy situation has gone on for long enough, ZX81 owners are banding together. There is unrest in the air. ZX81 owners are revolting! (O.K. who said that!?!)

Some of you may have noticed, in a certain computer publication, a rather alarming headline which read, "DEATH OF THE ZX81" in very large, frightening letters. Not a bit of it! Somebody extremely famous once said, 'The reports of my death have been greatly exaggerated', at a time when rumours that he had kicked the proverbial bucket were flying around like trapped flies in a jam jar. This sentiment very much applies to the ZX81, as neither the S.F.S.C. nor ZX Computing (they assure me) have any intention of letting this much loved machine hang up its micro processor and slip away to that happy computing land in the sky.

'How are we going to do that?' I hear you ask. The answer is *we're* not. YOU are! Even if you are not a member of the S.F.S.C. (shame on you!) you can contribute to this colume. Put pen to paper with any thoughts about the ZX81 on whatever subject that takes your fancy and send them to ZX81 CHATTERBOX at the Software Farm. The address is given at the bottom of the page. Ray Elder, the kindly editor of ZX Computing has said that if the response is good then ZX81 CHATTERBOX could be given more space in the future, so come on! With a bit of effort the ZX81 section could take over the whole magazine and from there – who knows? The world! The Universe! Clacton-on-Sea! (Where?) To give you an idea of what is expected here ae some examples of the sort of letters sent to the S.F.S.C.

Joystick Interface

At the moment the most frequent query the Cosmic Cockerel receives concerns the use of a joystick. Or to be a little more accurate, there seems to be a great deal of uncertainty as to the feasibility of connecting a joystick interface to the ZX81 at all. A typical letter reads something like this –

Dear Cosmic Cockerel
I have been thinking for some time about buying a joystick for my ZX81 in order to aid me in my perpetual battle against the electronic aliens. However, every time I think of it I look at my RAM pack sitting smugly on the only set of connections at the back, and it seems to me



that if I replace this with a joystick interface I will lose the use of the RAM pack and all my programs will be useless. Is there a way around this problem?
Yours sincerely, Worried of Worcester.

This is one of those problems which seem insurmountable, but the solution turns out to be so simple and obvious that it makes you want to crawl into a corner and beat yourself to death with the ZX81 manual! The connector on the joystick interface plugs into the computer, and the RAM pack in turn fits into the back of the interface. "Juss' like that!" as Tommy Cooper would have said.

There are two basic types of joystick interface available which vary only in the level of flexibility. With the lack of IN and OUT commands in ZX81 BASIC all the interfaces known to us mimic key depressions. This means that, to the computer, each of the four directions in which the joystick can be moved are seen as being identical to the pressing of one of four different keys. With the standard interface the cursor keys are usually used, with the addition of 0 for the fire button. This leads to one problem though. If a problem is written to be compatible with a joystick, those unfortunate souls without one find that they need double-jointed fingers in order to manually use the cursor key effectively. This can be overcome using software by including a 'define your own keys' facility such as the one we use in the Software Farm's hi-res programs. This allows the player to set the keys the game requires to either match an interface, or if none is available, to a pattern on the keyboard that is personally comfortable. Of course, not all programs carry this facility, so the other alternative is the use of a programmable interface but these are much more expensive. Instead of altering the program to suit the joystick, when using programmable interface, the key depressions mimicked can be altered to suit the program. If Mohammed can't go to the mountain. . . !

Both these types of interface are still available for the ZX81 from —

AGF Hardware
26 Van Gogh Place
Bognor Regis
PO22 9BY

On the other hand, in the current issue of the S.F.S.C. Newsletter there are instructions for the more ambitious on how to construct your own joystick and interface at a fraction of the cost of sending a monkey to the moon!

Machine Code!

Right! This is it! I am going to make publishing history in order to air a personal grievance. I'm going to have a good old go at you lot! By all means ask questions concerning machine code, it is the natural progression for someone wanting to know more about computers to want to delve into the mysteries of machine language. But, remember that it is a complicated subject and hardly likely to be understood overnight. We get lots and lots and lots and lots (and lots) of letters saying something like —

Dear Cosmic Cockerel
I am having a bit of bother learning machine code. Can you help?
Yours desperately, Dan
P.S. I am in a hurry. I am doing this for a school project and need to master it by the end of the week.

Now, what sort of a question is that?!! It's like saying you need to master Egyptian hieroglyphics because you are meeting King Tut a week on Tuesday! Or how about a quick degree in Brain Surgery after lunch, as your pet cat has told you of a desperate desire for a lobotomy? Or perhaps mastering a very long and nicely complicated piano concerto from scatch, in a single afternoon would suit you better?

(Doctor I really do feel better now that I've got that off my chest. Honest I do! Look aren't you putting this jacket on the wrong way round? Do you realise I can't move my arms? Doctor, come back! At least change the padding in this cell. It's a horrible colour. Doctor!. . .)

(We apologize for this interruption, sanity will be resumed as soon as possible — Ed.)

A little more seriously, a general pattern emerges from many of your letters on the subject of machine code. There seems to be a gap where the description of machine code commands ends, in the many books on the subject, and any explanation of how they can be of any practical use. It is all very well, you say, to be able to load the accumulator with the number of days in January, PUSH your date of birth onto the machine stack and take away the number you first thought of. But just how does this bear any resemblance to making hostile aliens attack an innocent, peace loving, but heavily armed fantasy world on the screen? Having just raved on about instant labotomies, obviously the answer is that there is not enough room here to explain. But, guess who comes to the rescue?

You got it! The Cosmic Cockerel on his gleaming white charger! In the S.F.S.C. Newsletter there is a machine code section with a difference. Using the age-old idea that one learns best by experience, over a number of issues a complete machine code program is written step-by-step; with full explanations of each stage as it develops. obviously, this demonstrates each machine code command, (or group of commands) in both the factual and practical sense. Good eh?

A Jolly Good Program

As a reward for wading this far through all the ramblings of a demented ZX81 programmer, and just to demonstrate that we do something sensible on rare occasions, here is a little present for you. Sent into the S.F.S.C. by Paddy Moindrot, it will both amaze and deafen your friends, by turning your ZX81 into a miniature organ. Yes! *Sound* through the TV set from your otherwise mute little friend! Not only that, but if you fancy mutilating (*or photocopying — Ed*) your copy of ZXC you can cut out the keyboard overlay and tape it over the keys. All you need now is the Vienna Boys Choir and you're well away!

The program covers two octaves in organ configuration. The top two rows being the higher octave and the bottom two the lower. You can BREAK out of the program by pressing SHIFT and SPACE together. Type it in exactly as listed, then RUN it (although it might be a good idea to SAVE it first!) A message will appear on the screen to tell you whether or not you have typed it in correctly. Assuming that you have, delete everything but the first line, add line 2 "RAND USR 16514" and you are ready to RUN the program proper. Ear plugs not supplied!

```
1   REM   88   DUMMY   CHARACTERS.....
..............................................................
..............................................................
10   LET   A$ = "CDBB02E5EB217FFCA7ED
52C8E1CBC4444D511428EBCDBD071134
401946AFB828DFDBFF783D20FDD3FF78
3D20FDC38240F5D6B9AB00E6C9009D00
6455473E00005B4D000000232E37001B
1F2731000006F84005F67788D78"

20 LET X=16514
30 IF A$ ="" THEN GOTO 90
40 LET B$ =A$( TO 2  )
50 LET A$ =A$(3 TO  )
60 POKE X,16 * CODE B$(1)+CODE B$(2)-476
70 LET X=X+1
80 GOTO 30
90 LET Y=0
100 FOR X=16514 TO 16604
110 LET Y=Y+PEEK X
120 NEXT X
130 IF Y=9734 AND PEEK 16511 =90 THEN
POKE 16510,0
140 IF Y=9734 AND PEEK 16511 =90 THEN
PRINT "ALL CORRECT — NOW DELETE ALL
  EXCEPT LINE 0":ND
150 PRINT"SOMETHING WRONG — CHECK ALL
  LISTINGS CAREFULLY"
```

If you wish to have details of the S.F.S.C. and want to join the ZX81 scene, or have any comment, query or anything to do with the ZX81 that could appear on this page then write to —

ZX CHATTERBOX
The Software Farm
16 Charlotte Street
Bristol
BS1 5PP



Readers' Reviews Revived!

As regular readers will know we recently dropped this feature, but some readers sent in these reviews and we couldn't resist them!



Middle Earth Trilogy
Level 9

To a seasoned adventurer 'Level 9' will have long been a household name, with games like *Lord of Time*, *Snowball* and *Return to Eden* to their credit. However Level 9 didn't make their name from these, excellent though they are. Any self-respecting adventurer worth his or her salt, really ought to have the Middle Earth Trilogy on his/her bookshelf. They are masterpieces of imagination, verbosity (some responses rival Infocom in length!) and speed. They really are the yardstick by which adventures should be measured. *Colossal Adventure*, *Adventure Quest* and *Dungeon Adventure* are well presented, supplied with a good instruction booklet (which gives VERY little away.) If you write to them they will send you a huge hint sheet, cleverly devised so that you don't accidentally find the answer to a puzzle you haven't yet come to.

The adventures are all text only, which will please many people who, like me, think that graphics are pretty, superficial and rather gimmicky. After all, if you've seen a static picture once, you don't really need to see it time and time again, particularly when it resides in memory and wastes valuable RAM. The locations (200 + in all three adventures) are 'graphically' described in eloquent text. They are all, of course, entirely machine coded, and Level 9 use a purpose built text compressor called 'a-code' which replaces text with signs in memory, and then reconstitutes it on screen. Thus a quart is very effectively squeezed out of a pint pot.

The puzzles in all the games

are clever, (reasonably) logical, and artfully devised to keep you coming back. Many's the time I've had to give up at two o'clock in the morning. 'slept on it' and come back the next day to solve a puzzle that kept me awake hours. There is an enormous list of objects, some useful and some valuable, to be collected on your journeys, and the answer to a puzzle in the last-but-one location! You really have to use your brain the whole time. In these games there is no such thing as luck! Well, not much.

Colossal Adventure is a faithful, even better, rendition of the original Crowther and Woods 'Colossal Caves' with the added bonus of an extra 70 location end-game. It is worth noting, however, that if you already have a copy of 'Classic Adventure' from Melbourne House, or indeed anything with a picnic area, bird and cage and PLUGH/XYZZY (!) then you'd better be careful not to duplicate a game you've already got, though I suspect that with the end game *Colossal Adventure* is a better buy.

The objective of the game is to enter the vast Colossal Caves, score maximum points by gathering up as much treasure as you can, rescuing some pitiful elves and living to tell the tale. The game is complex, but fairly easy to map. A word of warning: It isn't easy. As with all Level 9 games, response time is virtually nil, the vocabulary large, and text prolific to say the least.

Adventure Quest has a rather less vague objective: find the Dark Tower and destory the Demon Lord, Agaliarept, who resides therein. I only completed this game yesterday after a year and a half. The game starts off in the same area as *Colossal*, so those people who played *Colossal* will affectionately remember the little brick building etc, etc. However, after navigating the desert, nearly drowning in the underground river, exploring the underwater churchyard, meeting a Balrog on a rocky bridge over a huge chasm, and solving innumerable other tricks, traps and conundrums, you have far more to think about than mere sentimental memories! What is particularly frustrating with Quest is that you are only allowed to carry four objects at a time, so you often have to retrace your steps to collect objects you couldn't take the first time. This is really excruciating in the desert, but I suppose it's the price you have to pay for such a huge amount of objects.

And now, the pièce de résistance! *Dungeon Adventure* has no less than 100 puzzles to solve, over 220 locations and about 100 objects. Even the most experienced adventurer can expect to get fried a few times here, and end up like a stinking chip! Still, rats don't like stinking chips, so you should be all right.

The adventure takes place immediately after *Quest*, although you don't need to have played *Quest* to understand what's going on in *Dungeon*. It is well documented, and, if you go to the right place in the adventure you can get even more precise instructions. Basically though, the object of the game is to ransack the shattered Black Tower, getting out (alive) with as much loot as you can. Level 9 don't hinder you by only letting you carry four items here. You have a rather cleverly designed packing case to carry everything in. It also stops you getting killed. If you thought the other two games were tough — beware. This is the ultimate text only fantasy adventure.

All the Level 9 adventures show that care has gone into the concept and design of the adventures. For instance, you are never killed outright (what is more frustrating than sudden death, back-to-the-beginning adventures?) You are always given three lives, and in *Dungeon*, you can, theoretically go on for ever. None of their adventures have ever crashed on me, though occasionally the text compressor gets it wrong. Have you ever heard of 'a grate as rusted as y'? Neither had I, until I played *Quest*. Still these are only minor quibbles, with what must be the best trilogy of adventures on the market at the moment.

Colossal Adventure, *Adventure Quest*, and *Dungeon Adventure* are available in many shops, or by mail order from:

Level 9 Computing
Dep. V
229 Hughenden Road
High Wycombe
Bucks. HP13 5PG

They cost £9.90 each, and are worth every penny. If anyone needs help, or a complete map of any of these three, please send 70p to Simon Hollands, Woodside Cottage, Graffham, Petworth, West Sussex. GU28 0PU.

Taspro
Seven Stars Publishing
£3.95

LOADing and using Tasword Two is one of the pleasures of life. You know that it is going to do you proud and you are confident that you can leave it to get on with its job while you concentrate on writing and polishing your immortal prose. Yet one of the advantages of Tasword is that you are free to modify it if you have a little skill in Basic or machine code. This is not to say that Tasword Two is inadequate in its original form. Far from it, but there are certain modifications which you can make to personalise it and bring the operation and output even nearer to your own idea of perfection. It is possible, for example, to include a word count or page numbering routine.

However, such modifications may be beyond the programming skills of the user. Never mind. All is not lost, because you can buy ready made programs to MERGE with Tasword. One of these is Taspro, which produces a more professional output on dot-matrix and daisywheel printers by putting equal spaces between words in right justified text. Tasword Two, like many word processors, normally justifies lines by inserting extra space between words, starting at the end of the line. This, while perfectly adequate for normal use, can output text which is noticeably sparse on the right of the page. The Taspro modification avoids this and gives improved readability and superior appearance.

The program works by calculating the total amount of space necessary to justify each line and then divides it up between each word by sending non-standard but equal-length spaces to the printer. The appearance on the screen is not altered. Dot-matrix printers are switched to bit image mode and a number of high resolution spaces are printed between the words. In the case of daisywheel printers, the spacing pitch is altered accordingly under software control. The result is extremely good, with the added advantage that lines are justified even when there are embedded control characters or when using proportional spacing. Neither of these produces accurate justification in the standard Tasword Two.

The program comes on tape and is, in most cases, easy to use. You merely LOAD Tasword, go into Basic, LOAD Taspro and follow the instructions to set it up for your interface/printer combination. You then reload the Tasword Basic, run the program and resave it using the "t" option in the main menu. The program arrives ready-configured for proportional mode on Epson type dot-matrix printers. The only time you might have any slight difficulty is if you wish to use proportional spacing on daisywheel printers. In this case there is a certain amount of work to do before you can gaze proudly on your first beautifully printed page.

However, these difficulties are exceptional and 10 or 12 pitch on daisywheels or Pica and Elite on dor-matrix are relatively simple to do. Slightly inconveniently, each different configuration of Tasword has to be saved separately because there isn't room to cover all the possibilites from inside one program. But to be fair, it is feasible to have a limited number of variations selected from within the Tasword Basic, provided that you do not wish to use proportional spacing.

Although Taspro does not use any extra memory and the size of the text file is not reduced, I feel that it might have been better to put the code in the area used by the help pages. There is plenty of room there for extra code even if you do lose the quick reference to the key functions. But we know all these by heart anyway, don't we? By using this area, Ray Eckersley, the program's writer, could have avoided a problem which may affect some people. He has chosen to put his code into the addresses used for the software which runs some Centronics interfaces. You are therefore obliged to use an interface which does not take any of the Spectrum RAM. In my case, I bought the excellent Kempston E and the results are so good that I am glad that I made the decision to do so.

Overall, this is a very worthwhile addition to Tasword Two. Although it may be necessary to do some experimenting when setting it up, the effort is worthwhile and the results nothing short of superb.

Taspro is priced at a very reasonable £3.95, and available from: 15 Gloucester Ave, London NW1.

C. Laycock ■

Mindplay

This issue we welcome Brian Robb to the pages of ZXC (and apologise for wrongly crediting last issue's column to Greg Turnbull!).

Mordon's Quest
Melbourne House
£6.95

The cassette inlay for Melbourne House's new adventure *Mordon's Quest* claims 'the Classic Adventure continues. . .' for, this new game is written by the author of their original *Classic Adventure*. The game is aspiring to high standards, whether it reaches those standards or becomes as popular as *Classic Adventure* remains to be seen.

The task set in this adventure is quite daunting — simply to save the entire universe from destruction! You find yourself in a mysterious and atmospheric house, and after wandering around it a bit Mordon puts in an appearance. He then gives a speech filling two screens, which tells you the aim of the adventure. Your quest is to retrieve all the lost components of Mordon's precious immortality machine and bring them to him. To fail in this quest means the destruction of the universe, how exactly this will come about Mordon isn't telling. After this he disappears as rapidly as he appeared.

The adventure contains over 150 locations, all described by several lines of text, with no graphics. The location descriptions are very informative and detailed, and laden with atmosphere. It is very easy to visualise the rooms in the house, even the rather gaudy purple bathroom!

Soon, though, you have to leave the house and venture into the mist filled country side. If you find your way through the fog and mist (and it is very easy to get lost!) you should eventually come to a jungle. This is where things start getting tricky, with encounters with Quicksand smelling like old socks, and a man-eating plant protected by a conservation order! Once past these obstacles you may come across a hut which has a sign over the door displaying the letters A.A. and a plate and cutlery. Be warned — this may not be what you think!





In fact, in *Mordon's Quest*, nothing is ever exactly what you think. The adventure has the habit of throwing surprises at you from all directions, particularly once you reach the ruined city. As you move further into the adventure you seem to shift in time by thousands of years, ending up in a sparkling fantastic future world with perspex domes, flashing lights, and, of all things, ambient music! This clever twist of moving the player not only in space but also in time gives an interesting twist to what could have been another standard adventure.

Mordon's Quest is a very welcome throwback to the basics of a standard adventure game along the lines of *Classic Adventure*. It has none of the extra and often unnecessary frills which have become common in adventure games of late, such as memory wasting graphics, or highly interactive characters. Without these there is room for lengthier location descriptions which give this game an atmosphere missing from many more advanced adventures. Melbourne House, after the sophistication of *The Hobbit* and *Sherlock* have returned to basics for *Mordon's Quest*, and I believe it works well. That is not to say that this is a 'basic' game. It is a highly complicated adventure which will suit experienced adventurers who don't need pretty pictures to back up their imagination. The use of sophisticated text compression routines has made the puzzles in *Mordon's Quest* very challenging and complex, with very descriptive location text which gives the game a heavy atmosphere.

If you enjoy challenging, text only adventures, which may take several weeks to complete, then *Mordon's Quest* is for you. It really is an adventure in the

classic style.

Subsunk Firebird £2.50

The software arm of British Telecom has now moved into adventures with the inclusion of both *Subsunk* and *The Helm* in their range of budget software. Budget software does not always mean bad software and Subsunk proves this point being a very competent graphic adventure.

You play the part of a reporter from 'The Seafarers Gazette' with the highly unlilkely name of Ed Lines. You are on board the Sea Lion, a submarine, researching an article about life onboard a modern working sub. Life, though, is not being kind to Ed Lines, as the sub is attacked by an enemy fleet while he is carrying out his research. The sub is overrun by the enemy, but Ed escapes capture by hiding under a bunk. The enemy leave and sink the sub with Ed on board.

At the start of the game, you are trapped in the sub on the sea bed. Your objective is to find your way to the sub's telex room and send a message to bring a rescue party. The message you must send is 'subsunk'.

This is not as easy a task as you may suppose, as it's rather a large sub for you to find your way around, and several problems must be solved before you can send the message.

This is a Quilled text adventure with graphics. The graphics are fairly simple, though the majority are not static pictures. Many feature flashing consoles and even a sonar. There are also occasional sound effects which accompany the pictures. The text descriptions of locations are fairly simple as well, being just the name of the room, the exits and anything of interest. Unlike *Mordon's Quest*, which had no graphics, this game lacks atmosphere. Everything is very stark and sterile.

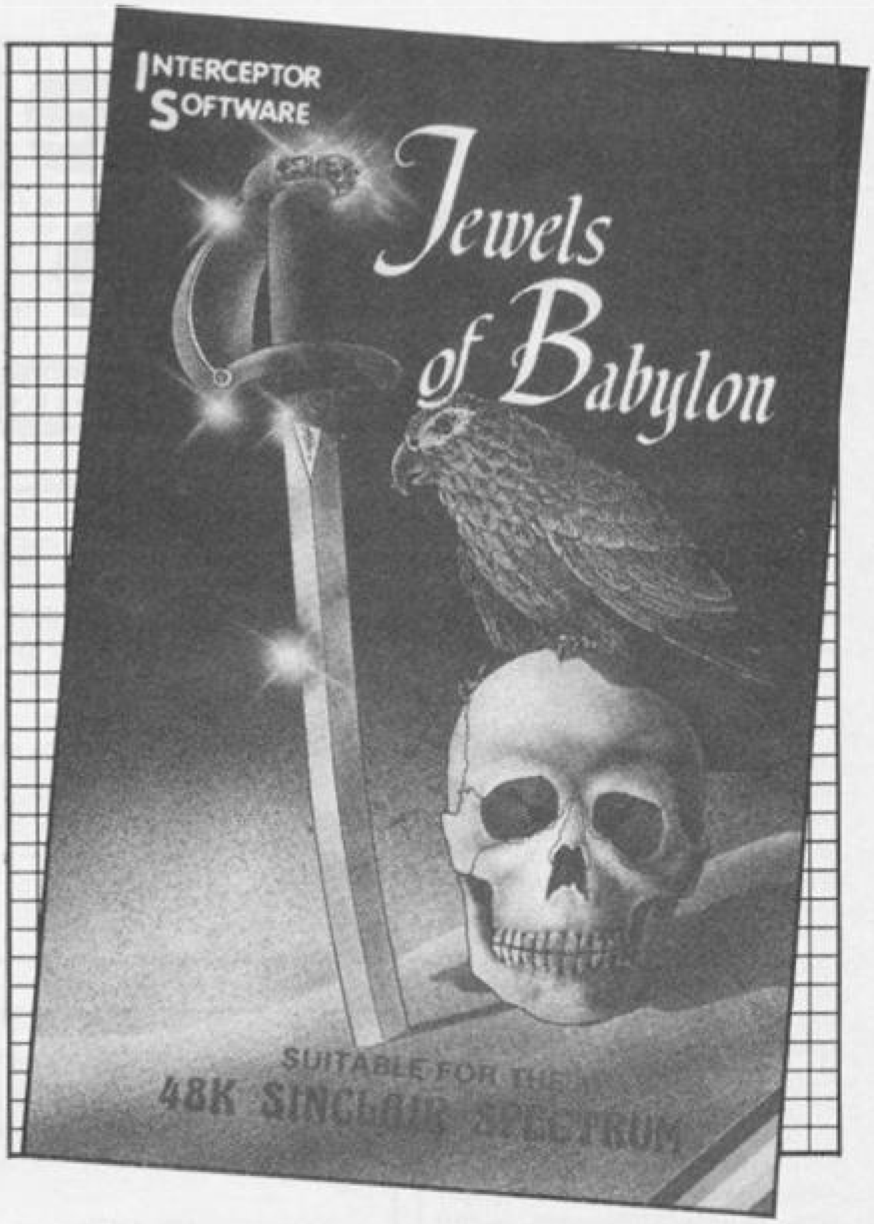
It is, though, a very enjoyable game, mainly because of its sense of humour. As with Imperial Software's *Clueso*, which I reviewed last issue, *Subsunk* rises above its status as a standard Quilled adventure because it is quite funny in places. For example, if you were to examine the sonar, the computer replies with 'You're on the sea bed, matey!' There is also the result of pressing a button marked 'Galley'. You smell a burning noise, and upon visiting the galley you are attacked by a burnt slice of toast which escapes from the exploding toaster!

There are incidents at almost all locations, and messing around with levers, buttons and switches will always produce a response of some kind, the majority of them in a humourous vein.

The game is adequately presented and laid out, though much more cannot be expected of a quilled game. The graphics are of a simple nature, and scroll off the screen as you type your responses. The game supports a limited vocabulary of around 150 words, but that is large enough for the scope of the game.

As an attempt to supply budget price adventures, *Subsunk* succeeds admirably, as there are worse games on the market for double the price. Firebird are continuing in the adventure market with *The Helm* at the same price. If you want a cheap, but challenging, adventure then *Subsunk* is for you.

Jewels of Babylon Interceptor Micros £5.50



The cover artwork for the cassette inlay of *Jewel Of Babylon* is strikingly similar to that for *Mordon's Quest*, as both feature skulls and swords. This is the only thing between the games which is similar because where *Mordon's Quest* is full of atmosphere with lengthy text descriptions, *Jewels Of Babylon* is not, despite the brilliant graphics which accompany certain locations.

The story goes thus: in Babylon, 3000 years ago, master craftsmen created a set of jewels, so beautiful that men would kill for them. By the end of the 19th Century, the jewels' bloody history had brought them into the possession of Queen Victoria, who intended to give them as a wedding gift to an Indian Princess. The ship carrying the jewels though, is attacked by pirates who leave the crew for dead and take the jewels. You have survived the attack and plan to regain the jewels. You travel to the pirate's island, which you must search to find the jewels, and return with them to your ship.

Several of the locations boast excellent graphic pictures, which give highly detailed views of your surroundings. The text descriptions though, are very sparse and don't adequately support the pictures. There are few taxing problems and it is possible to visit about half the game's locations with very little trouble at all. The adversaries for these first few locations are the island's wildlife — snakes, crocodiles, spiders and lions. Most of them are easily dealt with or easily avoided. There is very little challenge until you reach the later stages, despite the fact that the game claims to be for 'advanced players'.

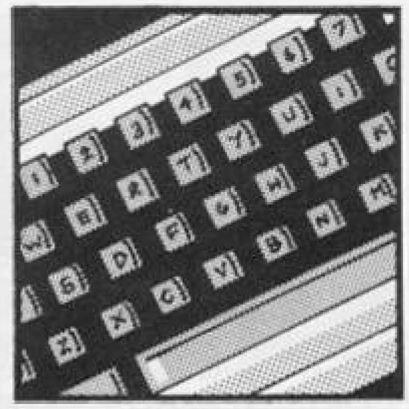
The game goes beyond the standard Verb/Noun input, but uses non-standard vocabulary which makes it more difficult to tell the game exactly what you wish to do. To leave your rowing boat, rather than CLIMB ASHORE or LEAVE BOAT, you have to enter CLIMB OUT BOAT. Small points like this increase the difficulty of the game, but in the wrong way. Rather than having difficult problems which need ingenious solutions, *Jewels Of Babylon* makes it difficult to find the correct phrases to use to get the response you want, which just results in annoying the player. Rather than playing a game, you're simply trying various phrases until you hit upon the right one. A more user friendly attitude would have improved the game.

Jewels Of Babylon is packaged in the video style cassette case which seems to be all the rage now with software houses. Fancy packaging does not make up for a poor game and only increases the price. If *Jewels Of Babylon* had been a £2.50 Firebird game then it would have been well worth the money.

Of the two graphic adventures reviewed this issue *Subsunk* and *Jewels Of Babylon*, *Subsunk* has simpler graphics but a far better game, whereas *Jewels Of Babylon* consists of excellent graphics but a very poor game.

Overall then, *Jewels Of Babylon* is a simple 'pirate and treaure' type game with spectacular graphics, but little game. If you want a graphic adventure then get *Subsunk*. If you want a REAL adventure, then get *Mordon's Quest*. Give *Jewels Of Babylon* a miss. ■

First steps in Machine Code



Part 6. by David Nowotnik

In this, the final part of our series on machine language programming for beginners, the remaining Z80 machine code instructions will be described. These deal with the index registers, with input and output commands, and with interrupts. All of these items have been left until now quite deliberately. The ideas involved in these commands are quite complex, and the beginners to machine language programming will probably manage quite happily without them. However, for completeness, these command sets will be covered. At the end of the article there will be a list of recommended books where these complex commands, and all the Z80 machine code instructions dealt with in the series can be found. And, in keeping with the general style of this series, there are some machine code programs which you can try for yourself. First of all though, let's examine the index registers.

Table 1. IN and OUT opcodes

IN A,(C)	ED78	OUT (C),A	ED79
IN B,(C)	ED40	OUT (C),B	ED41
IN C,(C)	ED48	OUT (C),C	ED49
IN D,(C)	ED50	OUT (C),D	ED51
IN E,(C)	ED58	OUT (C),E	ED59
IN H,(C)	ED60	OUT (C),H	ED61
IN L,(C)	ED68	OUT (C),L	ED69
IN A,(nn)	DBnn	OUT (nn),A	D3nn

Fig.1. ZX81 Verify program

a) Machine code loader routine

```
10 REM ..............    (at least 135 full stops)
20 LET X=16514
30 INPUT A$
40 IF A$="S" THE STOP
50 LET J=16*( CODE A$-28)+ CODE A$(2)-28
60 IF PEEK X=27 THEN POKE X,J
70 LET X=X+1
80 GOTO 30
```

b) Machine code bytes

```
CD 23 0F 37 11 00 00 CB 12 CB 0A CD 10 7C 18 FB 0E 01 06 00
3E 7F DB FE D3 FF 1F 30 49 17 17 38 28 10 F1 F1 BA D2 7A 7C
62 6B CD 10 7C CB 7A 79 20 03 BE 20 D6 23 17 30 F1 FD 34 15
21 09 40 50 CD 10 7C 00 CD 6C 7C 18 F6 D5 1E 94 06 1A 1D DB
FE 17 CB 7B 7B 38 F5 10 F5 D1 20 04 FE 56 30 B2 3F CB 11 30
AD C9 7A A7 28 BB CF 0C EB 21 7C 40 37 ED 52 30 06 1A B9 28
02 CF 1A 13 2A 14 40 37 ED 52 EB D0
```

c) BASIC routine to load machine code above RAMTOP

```
20 LET X=16514
30 FOR I=31744 TO 31878
40 POKE I, PEEK X
50 LET X=X+1
60 NEXT I
70 NEW
```

Index Registers

The Z80 CPU has two 16-bit registers, denoted IX and IY which perform as index registers. Unlike all the other so-called 16-bit registers, IX and IY only work in 16 bits, and with them you do not have the option of working with one half of the 'pair', i.e. 8-bits, at one time. IX an IY can be likened to the HL register pair, and many of the operations of IX and IY are identical to HL.

Those operations which work on IX and IY in the same way as HL result from direct addressing. In other words, operations carried out directly on IX and IY. For example, LD IX,nn and ADD IY,BC work exactly the same way as their counterparts with HL (LD HL,nn and ADD HL,BC).

What is special about IX and IY arises from their use in indirect addressing. Remember in an previous part of this series, (HL) denoted 'the byte in

Fig.2. Assembly language listing of ZX81 Verify

```
         CALL 3875          CD230F
         SCF                37
         LD DE,0            110000     Indicates no program name
         RL D               CB12
         RRC D              CB0A
loop     CALL listen        CD107C     Routine to receive signal
         JR loop            18FB         from tape.
listen   LD C,1             0E01
         LD B,0             0600
loop1    LD A,127           3E7F
         IN A,(254)         DBFE       Listen to tape
         OUT (255),A        D3FF      'Reflect' signal to screen
         RRA                1F         Bit 0 to carry
         JR NC to break     3049       Jump if BREAK pressed
         RLA                17         Bit 6 to carry to check
         RLA                17          if signal found.
         JR C to read       3828       Jump to routine to build
                                          up a byte.
         POP AF             F1         Remove return address
                                          from the stack.
         CP D               BA
loop5    JP NC to endch     D27A7C     Check if done
         LD H,D             62         Transfer DE (=0) to HL
         LD L,E             6B
         CALL listen        CD107C
         BIT 7,C            CB7A       The 'check name' routine
         LD A,C             79           from ROM 'LOAD'; not
         JR NZ,+3           2003         used by this routine.
         CP (HL)            BE
         JR NZ,-42          20D6
         INC HL             23
         RLA                17
         JR NC,-15          30F1
         INC (IY+15)        FD3415
         LD HL,16393        210940     First byte off tape is
loop2    LD D,B             50          address 16393.
         CALL listen        CD107C
         NOP                00
         CALL sysvar        CD6C7C     Check is system variable
         JR to loop2        18F6       Yes - continue listening.
read     PUSH DE            D5
loop3    LD E,94            1E94       Set timing loop for tape
         LD B,26            061A
loop4    DEC E              1D
         IN A,(254)         DBFE       Tape read
         RLA                17
         BIT 7,E            CB7B
         LD A,E             7B
         JR C to loop3      38F5
         DJNZ to loop4      10F5
         POP DE             D1
         JR NZ to done?     2004         Is the byte complete in
         CP 86              FE56           register E?
         JR NC to listen    30B2         Continue listening....
         CCF                3F
         RL C               CB11
         JR NC to listen    30AD          for more bits.
         RET                C9           Byte complete
break    LD A,D             7A           If BREAK not pressed...
         AND A              A7
         JR Z to loop5      28BB          carry on listening.
         RST 08             CF0C         Otherwise give error 'D'
```

memory whose address is the value contained in the HL register pair'. This is indirect addressing. IX and IY can also be used for indirect addressing, but their use differs from that of HL. Instead of simply having (IX) or (IY), the assembly language mnemonics for indirect addressing with these registers are (IX + d) and (IY + d). 'd' stands for displacement. It is a one byte value in the operand (so having a value between 0 and 255) which is added to the value in IX or IY to obtain the effective address for indirect addressing. The value of this extra facility may not be immediately obvious, but it is often used by machine code programmers for sequentially accessing a block of bytes.

This extra facility offered by the index registers can be quite powerful, but is only likely to be used by the more experienced machine code programmer, so we'll deal with it in theory only. The similarity of IX and IY to HL continues in the same way the machine code bytes are formed. Take an example like ADD HL,BC. The machine code opcode for this is a single byte instruction, 09 hex. To get the corresponding machine code for IX (ADD IX,BC) you precede the byte 09 with DD hex, and for IY, the opcode is preceded with FD. This pattern governs the formation of all IX and IY opcodes; precede the opcode for the corresponding HL instruction with DD for IX and FD for IY. ADD IX-BC is DD09; ADD IY,DE is FD19; PUSH IX is DDE5 (PUSH HL is E5) and POP IY is FDE1 (POP HL is E1). Simple!

The formation of the indirect addressing instructions with IX and IY is also simple once you know the rules; again, DD and FD precede the corresponding HL opcode, but you must remember that there is at least one operand, the displacement 'd'. For example, DEC (HL) has the opcode 35. DEC (IX + d) has the opcode DD35, to which the value of 'd' must be added, to give a three byte instruction. If d is to be 8, then the three bytes for DEC (IX + 8) are DD3508 hex.

These rules get a little more complicated when there is a further operand. For example. LD (HL),20 (hex) has the opcode 36 and an operand 20, so the instruction is 3620. For LD (IY + 08),20 the opcode is FD36. Next comes the value of 'd' (08), and finally the operand 20, giving a four byte instruction FD360820. And, just to ▶

Figure 2 continued

```
sysvar   EX DE,HL        EB       Address into DE
         LD HL,16508     217C40   Start address of program
                                     area.
         SCF             37       Set carry
         SBC HL,DE       ED52     Is address still in
                                     system variables?
         JR NC to end?   3006     If not, jump
         LD A,(DE)       1A       Is the byte in RAM...
         CP C            B9         the same as on tape?
         JR to end?      2802     If so, then jump
         RST 08          CF1A      otherwise generate the
                                     error message 'R'
end?     INC DE          13       Next address
         LD HL,16404     2A1440   HL set to end of
                                     variables
         SCF             37
         SBC HL,DE       ED52     Has end been reached?
         EX DE,HL        EB
         RET NC          D0       If not, carry on
         RST 08          CF1E       if yes, give '0'
                                     end message.
```

Fig.3. Spectrum Byte Scroll routine – BASIC loader

```
10 CLEAR 29999
20 LET x=30000
30 READ a$: IF a$="s" THEN STOP
40 FOR i=1 TO LEN a$ STEP 2
50 LET y=0
60 FOR j=0 TO 1
70 LET r= CODE (a$(i+j))
80 LET s=r-48-(r>57)*7
90 LET y=y+16*s*(j=0)+s*(j=1)
100 NEXT j
110 POKE x,y: LET x=x+1
120 NEXT i: GOTO 30
200 DATA "21FF5701C000C501E1FF097E54"
210 DATA "5D23011F00EDB02B7701E0FF09C1"
220 DATA "0B78B120E60118001100582101 58"
230 DATA "1AC5011F00EDB02B77132323C10B"
240 DATA "78B120EEC9"
250 DATA "21004001C000C5011F00097E54"
260 DATA "5D2B011F00EDB823770120 0009"
270 DATA "C10B78B120E6011B0011FF5A21FE"
280 DATA "5A1AC5011F00EDB823771B2B2BC1"
290 DATA "0B78B120EEC9"
300 DATA "s"
```

confuse you a little more (*help! – Ed's asst.*), the rules are yet more complicated for the formation of indirect addressed instructions of the BIT,SET, and RESET of IX and IY. Take BIT 6 (HL); the opcode for this is CB76. BIT 6, (IX+08) is DDCB0876; in other words the corresponding HL opcode is now split when translated to the IX or IY opcode by the value of 'd'.

These fiddly rules emphasise the usefulness of assemblers, which convert assembly language mnemonics to machine code. It is so easy to forget how the instruction is made up, that mistakes in manual coding of IX and IY instructions are common. Indexed addressing with the Z80 CPU is not as powerful as with other processors, such as the 6502, so if you don't want to use them, it is usually relatively easy to just use the other registers instead. One good reason for avoiding the use of IX and IY is that they are used by the BASIC interpreters and operating systems of the ZX81 and Spectrum, so altering their values could cause a crash on return to BASIC (or with SLOW mode on the ZX81).

In/Out

The IN and OUT instructions enable the Z80 to communicate with the 'outside world'. So far, the machine code instructions have only dealt with operations within the CPU itself, or communication between the CPU and memory. For effective operation, the Z80 must be able to communicate with other devices; notably the keyboard, the cassette interface, and (if you have one) the printer. In this list of external devices, the TV or monitor isn't always included. As we have seen before, the screen display occupies a certain area of RAM, and this is translated to a screen image through specialised hardware and software connections.

The hardware of the computer assigns to these external devices an identifying number, and connects them to the CPU through links which are termed 'ports'. The IN and OUT instructions allow values (bytes) to be transferred between the CPU and these ports. IN transfers a value from the port to the Z80, while OUT enables the opposite transfer to occur.

There are two types of IN and OUT instruction, the simplest are IN A,(C) and OUT A,(C). The brackets around 'C', if you recall, suggest indirect addressing. In this case it means that the register 'C' holds the identifying value of the port. So, IN A,(C), with 'C' holding a value of 254 would instruct the transfer of a value from port number 254, and place it in register 'A'. Similarly, OUT (C),A sends the value in A to port 254. Similar instructions are available for all the registers, and their opcodes are listed in table 1. They are all two byte opcodes, preceded by ED.

Also two bytes in length are the outer IN and OUT instructions, but in these cases the opcode is only one byte. These take the form IN A,(n) and OUT (n),A. Here, transfer of data is only possible to and from the 'A' register. The port is identified by the operand which follows the single byte opcode; DB for IN and D3 for out. Hence the instruction DBFE collects a value from port 254, and places it in register 'A'. Port 254 is the cassette interface on both the ZX81 and Spectrum. You could write your own LOAD and SAVE routines in machine code knowing that fact; but, if you did want to use these routines in machine code, it is far easier to use the routines already available in ROM for performing these functions.

One routine the ZX81 does lack is the VERIFY facility, which checks that a program on tape matches the program in the machine (to ensure proper SA-

Fig.4. Assembly language listing of the Spectrum byte scroll routine.

```
(Left)  LD HL,22527     21FF57   Last screen address
        LD BC,192       01C000   number of lines
loop    PUSH BC         C5       Save this value
        LD BC,-31       01E1FF   Calculate the first
        ADD HL,BC       09        address of the line.
        LD A,(HL)       7E       Save byte in 'A'
        LD D,H          54       Put address into DE
        LD E,L          5D
        INC HL          23       Next address
        LD BC,31        011F00   32 columns -1
        LDIR            EDB0     scroll left
        DEC HL          2B       Set to column 32
        LD (HL),A       77       wrap-around
        LD BC,-32       01E0FF   Set HL to start
        ADD HL,BC       09        of the next line.
        POP BC          C1       Recover the line
                                   counter.
        DEC BC          0B       Decrement counter
        LD A,B          78       Check if zero
        OR C            B1
        JR NZ to loop   20E6     Continue if not zero
        LD BC,24        011800   24 Attribute lines
        LD DE,22528     110058   First attribute address
        LD HL,22529     210158  Second attribute address
loop2   LD A,(DE)       1A       Store attribute
        PUSH BC         C5       Store line counter
        LD BC,31        011F00
        LDIR            EDB0     Scroll left
        DEC HL          2B       Set HL to last byte
        LD (HL),A       77       wrap-around
        INC DE          13
        INC HL          23       Set HL and DE to
        INC HL          23         the next line.
        POP BC          C1       Recover line counter
        DEC BC          0B       Decrement counter
        LD A,B          78       Check if zero
        OR C            B1
        JR NZ to loop2  20EE     Go back if not zero
        RET             C9       Else return to BASIC.
```

The scroll right routine is very similar to the scroll left program, so it will not be repeated here.

VEing of program). This routine is given as one of the examples in the ZX81 section. It uses the IN A,(254) command to read a byte from tape. You'll also notice a command OUT (255),A — this causes the wiggly lines you see on the screen when loading by 'reflecting' the byte value read from tape (in 'A') onto the screen, which, in this case, has been configured as port 255.

There are block IN and block OUT instructions, which are equivalent to LDIR and LDDR with load, but these are rarely used.

Finally, let's take a look at interrupts.

Interrupts

Interrupts are a special form of communication for the CPU. Whenever you are working with the computer, it seems that the CPU seems to progress through machine code (either yours, or that in ROM when you are working in BASIC) continually. However, the CPU is often being interrupted. An interrupt is caused by a device connected to the CPU which demands priority over the current work of the CPU. Let's first take a look at the various types of interrupt, then at some examples to see the purpose of interrupts.

Although there are other types of interrupt available to the Z80 CPU, the important ones for the ZX81 and Spectrum and the non-maskable interrupt (NMI) and maskable interrupt. As their names suggest, you cannot prevent an NMI, but you can stop a maskable interrupt.

In all forms of interrupt, a signal is sent to the CPU via an interrupt line, by a device requesting service. A good example of an NMI is the screen refresh on the ZX81. To keep down the cost of the ZX81, it was designed with the screen refresh routines in ROM. The screen refresh causes the TV image to be updated every 1/50th of a second. Every 1/50th of a second, an NMI is generated (by an internal clock) which tells the CPU to stop whatever it is doing. With an NMI, the current value of the program counter (PC) is stored (so that the current program can continue after the interrupt), and the value of 66 hex placed in the PC. All NMI routines start at this address. In the ZX81, this is the start of the screen refresh routine. One of the first actions of this routine is to check whether SLOW or FAST mode is set. If FAST is set, the NMI is terminated; if SLOW is set, then the screen is refreshed. As the ZX81 works four times faster in FAST than in SLOW mode, you can see how long and complex it is to renew the screen image. The NMI routine on the Spectrum does little, apart from resetting the machine (RST 00).

NMIs are of little use to the programmer, as they cannot be controlled. This is not the case with maskable interrupts. These interrupts can be masked by the programmer with the machine code instruction DI (Disable Interrupt) and permitted with EI (Enable Interrupt). There are three types of maskable interrupt denoted by mode 0,1, or 2. These are set by the machine code instructions IM 0, IM 1, and IM 2. So, whenever a maskable interrupt signal is received by the CPU, it is only accepted if the interrupt is not masked, and then the appropriate interrupt mode enacted, depending on which mode is set by the programmer (or by ROM, when in BASIC).

Mode 0 has a quite specialist use. It requires that the device requesting the interrupt places, on the data bus, the machine code bytes to be interrupted by the CPU. For example, a ROM or EPROM overlaying the main ROM could be brought into use by this interrupt.

Mode 1 saves the PC contents on the stack, and places the value 38 hex in the PC. The address is the start of a ROM routine, which on both the ZX81 and Spectrum causes the keyboard to be read. In normal operation, these computers have interrupt enabled, and mode 1 set, which allows the keyboard to be scanned for a keypress every 1/50th of a second. Sometimes, however, it is not convenient for this continual scanning to take place. The ▶

LOAD and SAVE routines are one example. At certain times in the LOAD and SAVE sequence, it is vitally important that the listening or output routines are not interrupted. During this period, interrupts are disabled. During non-critical periods in the LOAD or SAVE routine, interrupts are enabled, to allow the keyboard to be scanned (to check for the passing of BREAK).

Mode 1 is a fixed routine. While it is useful for the programmer to know what it does, he can do little with it apart from turning the keyboard scan routine on or off.

Mode 2 provides a lot more flexibility. It makes use of another register, the I (or interrupt) register. When a mode 2 interrupt is encountered, the PC contents are stored as with the other interrupts, and PC is filled with two values. One is supplied by the device calling the interrupt, the other is in the I register. Together they form a 16-bit address, which points to another two byte value held in memory. It is this second address which is placed in PC. This 'vectored' system allows the user to define where the interrupt handling routine is, so allowing the user to write his/her own interrupt routine, and have, if desired, a table if address vectors. The programmer can set the value of the interrupt register with the command LD I,A. This interrupt is something for the experienced programmer, but is still rather useful.

Some Good Books

O.K. That's all the theory covered! If the series has given you an appetite for machine code, then you may well want to take your studies further. An excellent 'bible' for Z80 machine code programmers is the book by Rodnay Zaks, 'Programming the Z80'. This covers all the theory of Z80 programming in great detail, and is an excellent reference text. If you want something a little bit more machine specific, then Toni Baker's book 'Mastering Machine Code on your ZX81' and '40 Best Machine Code Routines for the ZX Spectrum' by John Hardman and Andrew Hewson can both be recommended. And, of course, ZX Computing frequently carries some complex machine code articles to get your teeth into!

To finish off, here are two machine specific examples which incorporate many of the principles covered in the series. For the ZX81 there is a VERIFY routine; for the Spectrum a left/right byte SCROLL which includes screen display and attributes, and a 'wrap-around' effect.

ZX81 'VERIFY' Routine

VERIFY is a command which was excluded from the ZX81 due to the lack of space in the ROM, yet it is one for which users have a great need. LOADing and SAVEing tend to be unreliable on the ZX81, and there is no convenient way of checking that a program is successfully saved before pulling the plug, and losing the lot! However, the following piece of machine code performs the verify operation. Having saved a program to tape, the verify routine will check that the routine on tape is a true record of the program in ROM.

This verify routine, to be honest, is based almost entirely on the LOAD routine in the ZX81's ROM. The only change is at the vital point when a byte is read off tape. Instead of putting that byte into the appropriate place in RAM, it is compared with the current byte at that address. If there is a match, then the routine continues to check the next and subsequent bytes. If there is a mis-match, then the routine exits with an error message (R/O). If the verification was successful, with no mis-matches, then the 0/0 message will be returned at the end of the routine.

The machine code loading routine is shown in fig. 1. For convenience the routine is held in a REM statement (135 characters long), so that it can be saved onto tape. So, first type in the long REM line (line 10) and the loading routine fig. 1 (a), then type in the hex digits, one at a time, reading from left to right, given in fig. 1 (b). Then delete all lines except line 10, and type in the lines given in fig. 1 (c). These will transfer the machine code to above RAMTOP when required. Finally, save the REM line and loading routine onto tape.

To use the verify routine, it must be loaded into your 16K ZX81 prior to any other program. This is the sequence of actions. First, lower RAMTOP with the two direct commands:

POKE 16388,123
POKE 16389,255



'We're in trouble. The ZX81's crashed again!'

This isolates just enough space at the top of RAM for the verify routine. Now, lower the stack, by using NEW as a direct command, LOAD and RUN the verify program; this installs the machine code above RAMTOP. Finally, NEW the machine code loader, and start typing in your progam.

After you have SAVEd your BASIC program onto tape, and you want to verify your recording, rewind the tape to the start of the program in tape, type in the direct command

RAND USR 31744

Press play on your tape recorder, and NEWLINE on the computer. The screen will appear as if the program were being loaded. There is no check of the name of the program on tape, so be sure to play back the right program (if you have more than one on the same tape). If you saved the program with variables, then CLEARed them before verifying, or changed the variables in any way, then you may get a verify error. Otherwise, if all is well, you'll get an 0/0 message to tell you that the program has been verified.

The assembly language listing of the verify routine appears in fig. 2.

Spectrum Scrolling Routines

The final Spectrum example in this series is one for special effects. It allows full screen scrolling, left or right, moving attributes as well, and incorporates wrap-around. This effect means that characters which fall off the edge of the screen re-appear on the other side. If you don't want the wrap-around effect, you can replace the bytes underlined in the hex dump of fig. 3. with zeros (00).

So, first type in the hex loader in fig. 3., then RUN it to install the bytes. The code is fully relocatable (i.e. it doesn't matter where in RAM you store it, it should always work), so on a 48K machine, you could load it higher in RAM, so that not so much memory is wasted.

Once you run the program, the machine code is placed above RAMTOP, and you can NEW the loader. The direct command:

RANDOMISE USR 30000

will cause the whole screen display to move one byte to the left, while:

RANDOMISE USR 30060

causes the display to move to the right. if you want to incorporate these routines in a BASIC program, then one suggestion is to have the display move when the appropriate 'arrow' key is pressed; for example:

```
200 IF INKEY$ = "5" THEN
RANDOMISE USR 30000
210 IF INKEY$ = "8" THEN
RANDOMISE USR 30060
```

The assembly language listing for these routines appears in fig. 4.

Well — that's it! I hope you have enjoyed reading the series as much as I've enjoyed writing it. You should find machine code much simpler than you ever expected. Try writing your own routines, and carry on working out how routines written by others do work — there will always be routines in ZX Computing for you to try. Good luck!

Tortoise Wise

Even more lines from a Parent Who Gets Left Behind — By David Stewart

The game's up. . the cat's out of the bag. . . . the balloon has gone up. My two sons now know that their father and the author of this little column in ZX Computing every issue are one and the same person. It's hard enough to keep a secret in our house, so I should have known better I suppose. They just never guessed that it was me and well, I had my reasons for not telling them. I just has no way of guessing that their reaction would be so extreme.

'Look upon it as a kind of compliment,' I reasoned. 'You come out of it quite well.'

'Tortoises and Hares,' they spluttered in such a way that somehow it sounded slightly unpleasant. 'You've really done it this time Dad.'

I heard them storm up to their rooms and then listened to the odd verbal explosion that shook the windows every few minutes for the next half hour, while I imagine they reread back copies of the magazine in a new light. Even their footsteps when they came back down again were angry.

'You have taken advantage of us,' the eldest announced. 'We demand compensation.'

'And another Spectrum for US,' his brother adds. Then as an afterthought, 'make that Two.'

I try to explain to the stony faces and the folded arms that I do not share their point of view. That if they could just calm down they might see it all my way. But it was a waste of time because for children and Hares alike there is nothing quite so painful as injured dignity. And they believed that by not being consulted in the first place they had been betrayed.

They didn't speak to me for several days. but each time I went in to work on the computer the strain of knowing what I was doing or if I was writing about them became too much. In the end we sorted things out and the wallet didn't take much of a pounding. They agreed. We were just like the Tortoise and the Hare. I didn't tell them then, but of course they will soon find out, that I counted that as a victory — Tortoise Wise. . . . .

But, it did set me to wondering again. Surely I'm not the only one? There must be other parents who, like the Tortoise, get left behind by their kids. Others, who like myself find their micros harder to master than their children do, who take hours to learn what their children only seem to need minutes to understand, and for whom it is one long race just to keep in touch with a world our children are quite at home in. Computers. . . . .

So maybe it is time that we Slower Ones got in touch. If you have had the kind of experience I've related in this column over the months why not drop me a line? The more humorous the better. The one thing that keeps me going, Tortoise Wise, is seeing the funny side (though I'm not sure that that is true of the Hares). So, put pen to paper all you Tortoises out there and let's be hearing from you. . . . .
Tortoise Wise.

(Well it's over to you ZX readers, address letters to David Stuart, Tortoisewise, and send them to our usual address — Ed.)

Pluto Adventure

David Knight may not have shining armour, but the quest he sets from the safety of Didcot would challenge King Arthur's best!

In this adventure, which runs on a 48K ZX Spectrum, you play the part of Ferdo, a scientist researching into atomic energy.

The year is 2066, and a new civilisation has been discovered below the surface of the planet Pluto. The small community appears to be devoid of life. The Mekon, most wanted criminal on Earth, is known to have transported to the underground village in his deluxe spaceship. You and your fellow Earthman, Bill, have gone to the planet to trap him. You must find and pilot home the Mekon's ship, leaving him stranded on the planet. Other beings known to have transported in their ships to Pluto are: Gasher, an android from planet Mars; Pharto, a Noxian; Ilko, a subterranean from Mercury (he is used to being underground); and finally, Toronto, a Korzaxian from Korsak.

Bill, Pharto and Toronto, being friends of yours, will collect information to help you. The Mekon, Gasher and Ilko, however, are very likely to attack on sight. You can only control the Mekon's ship, the others being beyond your comprehension. Unfortunately, you cannot yourself take a ship with you. You use the matter-transporter which you developed. It cannot get you back!

Entering the Program

First, type in program 2. Save it at the beginning of a tape with: SAVE "Pluto Load" LINE 10. NExt, type in program 3. Save this after the other program, with: SAVE "Pluto" LINE 1. Now, type in program 1, run it, and SAVE the data from it after Program 3 with: SAVE "Data" DATA a( ): SAVE "Data 1" DATA o( ): SAVE "Data 2" DATA c( ): SAVE "Data 3" DATA c$: SAVE "Data 4" d$( ): SAVE "Data 5" DATA h$( ): SAVE "Data 6" DATA o$( ). Now, if all of the data and program is correct, you can LOAD it from tape by typing LOAD "" or LOAD "Pluto Load" and playing the tape from the beginning. Instructions and vocabulary are given in the program.

Variables

Arrays

a	contains the data for the map.
o	contains the data for the objects.
c	contains data for the characters.
c$	contains names of objects and characters.
d$	contains 53 descriptions of positions.
o$	contains object descriptions.
h$	contains character descriptions.

Strings

s$	name for Saving and Loading.
z$	the string the user types in.

Variables

Bill1, gasher1, toronto1, pharto1, mekon1, ilko1
Variables used by the computer to check whether a character is in the same place as Ferdo.

Ferdo, bill, gasher, toronto, pharto, mekon, ilko
Variables which point to subroutines.

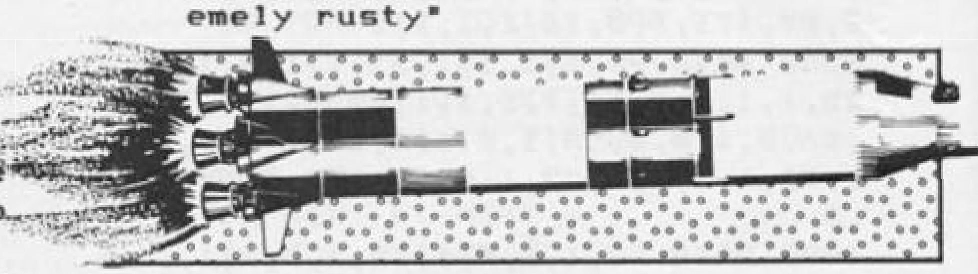
x	multipurpose variable
p,q	co-ords of characters (consistently change)
yy	a check for whether there are any objects
k	a number for different objects
r,r1	random numbers for fight sequence.

Program 1

```
10 DIM a(8,15,5)
20 DIM o(7,5)
30 DIM c(7,7)
40 DIM c$(2,7,7)
50 DIM d$(53,64)
60 DIM h$(6,11)
70 DIM o$(7,32)
80 FOR a=1 TO 15: FOR b=1 TO 8: FOR c=1 TO 5: READ a(b,a,c): NEXT c: NEXT b: NEXT a
90 FOR a=1 TO 7: FOR b=1 TO 5: READ o(a,b)
100 NEXT b: NEXT a
110 FOR a=1 TO 7: FOR b=1 TO 7: READ c(a,b): NEXT b: NEXT a
120 FOR a=1 TO 2: FOR b=1 TO 7: READ c$(a,b): NEXT b: NEXT a
130 FOR a=1 TO 53: READ d$(a): NEXT a
140 FOR a=1 TO 6: READ h$(a): NEXT a
150 FOR a=1 TO 7: READ o$(a): NEXT a
```

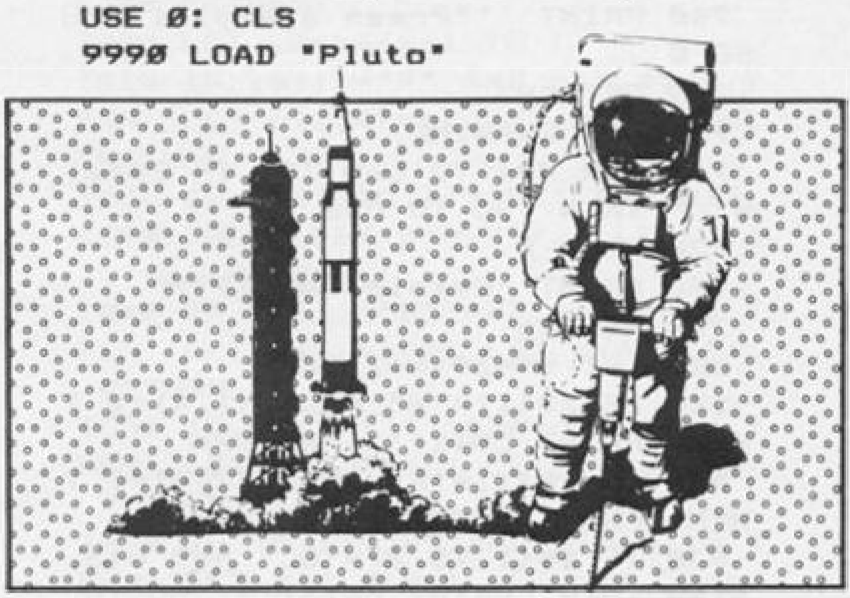
```
1000 REM a
1010 DATA 1,0,1,1,0,1,0,0,0,1,2,
0,0,1,0,3,0,1,1,1,4,0,0,1,1,5,0,
1,1,1,6,0,0,0,1,7,0,1,0,0,1,1,1,
0,0,8,0,1,0,0,7,0,1,0,0,9,1,1,0,
0,7,0,1,0,0,10,1,0,0,0,7,0,1,1,0
,7,1,1,0,1,11,1,1,1,0,12,1,1,0,1
,7,1,0,1,0,13,1,1,1,1,7,1,0,1,1,
7,0,0,1,1,7,1,0,1,1,7,1,0,0,1
1020 DATA 14,1,1,1,0,15,1,1,1,1,
16,0,1,1,1,17,1,1,1,1,18,1,1,1,1
,14,0,1,1,1,19,0,1,1,1,20,0,1,0,
1,21,1,1,1,0,22,1,1,1,1,23,1,1,1
,1,24,1,1,1,1,25,1,1,1,1,26,0,0,
0,0,27,1,1,1,1,28,1,1,0,1,29,1,0
,1,0,29,1,0,1,1,30,1,0,1,1,31,1,
1,1,1,31,1,1,1,1
1030 DATA 32,1,1,1,1,29,1,1,1,1,
29,1,1,0,1,33,0,1,1,0,33,0,1,1,1
,33,0,1,1,1,34,1,1,0,0,31,1,1,1,
0,35,1,1,1,1
1040 DATA 36,1,1,1,1,36,1,1,0,1,
31,1,1,1,0,31,1,1,1,1,31,1,0,1,1
,31,1,1,0,1,25,1,1,1,0,37,1,1,1,
1,36,1,1,0,1,36,1,1,0,0,31,1,1,0
,0,38,1,0,0,0,39,0,1,1,0,33,1,1,
0,1,31,1,1,1,0,40,1,1,1,1,36,1,1
,1,1,36,1,1,0,1
1050 DATA 31,1,1,0,0,38,0,0,1,0,
41,1,1,1,1,33,1,1,0,1,31,1,1,1,0
,31,1,1,1,1,42,1,0,1,1,43,1,0,0,
1,31,1,1,0,0,41,0,1,1,0,41,1,1,1
,1,33,1,1,0,1
1060 DATA 25,1,1,1,0,31,1,1,0,1,
44,0,1,1,0,44,0,0,0,1,16,1,0,1,0
,45,1,1,1,1,41,1,1,1,1,33,1,1,0,
1,46,1,1,1,0,31,1,1,1,1,44,1,1,1
,1,47,0,0,0,1
1070 DATA 27,0,1,1,0,41,1,1,1,1,
37,1,1,1,1,33,1,1,0,1,20,1,1,1,0
,48,1,1,0,1,44,1,0,1,0,44,0,0,0,
1,29,1,1,1,0,16,1,1,1,1
1080 DATA 20,1,1,1,1,33,1,1,0,1,
25,1,1,1,0,16,1,1,1,1,49,0,1,1,1
,16,0,1,0,1,50,1,0,1,0,51,1,0,1,
1,51,1,0,1,1,52,1,0,0,1,53,1,0,1
,0
1090 DATA 12,1,0,1,1,12,1,0,1,1,
20,1,0,0,1
2000 REM o
2010 DATA 1,1,2,1,10,2,2,2,8,15,
3,3,3,1,15,4,4,30,4,13,5,5,9,8,1
0,6,6,11,4,13,7,7,8,1,1
3000 REM c
3010 DATA 1,1,10,15,4,9,1,2,1,8,
10,1,5,1,3,2,14,4,5,15,-1,4,3,5,
9,8,12,1,5,4,10,17,6,9,1,6,5,7,1
6,7,1,-1,7,6,9,5,3,1,-1
4000 REM c$
4010 DATA "Ring","Key","Scroll",
"Boat","Torch","Potion","Sword",
"Ferdo","Bill","Gasher","Toronto
","Pharto","Mekon","Ilko"
5000 REM d$
5010 DATA "You are in the ruins.
"+CHR$ 13+"All around you, build
ings"+CHR$ 13+"collapse"
5015 DATA "You are in the dining
 room."+CHR$ 13+"You see Ilko's
rocket."+CHR$ 13+"It is empty."
5020 DATA "You're in the hall."+
CHR$ 13+"A door is to the south.
"+CHR$ 13+"There is a table."
5025 DATA "You are on a staircas
e."+CHR$ 13+"There are walls to
the north andsouth."
5030 DATA "You are in a corridor
."+CHR$ 13+"Open doors are to th
e East and"+CHR$ 13+"South."
5035 DATA "You are in the King's
 bedroom."+CHR$ 13+"The Mekon's
ship is here."
5040 DATA "You're in the castle
gardens."+CHR$ 13+"There are nea
t hedges"+CHR$ 13+"everywhere."
5045 DATA "You're in a grassy cl
earing"+CHR$ 13+"There are high
walls surroundingyou."
5050 DATA "You are at the Castle
 Door."+CHR$ 13+"There is a gold
 lock on the"+CHR$ 13+"door."
5055 DATA "You are in a bathroom
."+CHR$ 13+"There is a corridor
to the North"
5060 DATA "You are in the gateho
use."+CHR$ 13+"Doors to the West
 and South."
5065 DATA "You are in a clearing
 with a"+CHR$ 13+"pond"
5070 DATA "You are on the castle
 drive."+CHR$ 13+"The entrance i
s to the North."
5075 DATA "You are on a pile of
rubble."+CHR$ 13+"This place sti
nks!"
5080 DATA "You are in a clearing
 with a"+CHR$ 13+"tent. It is em
pty!"
5085 DATA "You have arrived at a
 clearing"+CHR$ 13+"with a tree
in it."
5090 DATA "You are at the castle
 gate."+CHR$ 13+"To the North is
 the castle."
```

```
5Ø95 DATA "You are at a statue."
+CHR$ 13+"It is of the king."
51ØØ DATA "You find a cauldron h
ere."+CHR$ 13+"It is of no use t
o you."
51Ø5 DATA "You are in a clearing
."
511Ø DATA "You are in a clearing
."+CHR$ 13+"You see Bill's rocke
t."+CHR$ 13+"It is empty."
5115 DATA "You are in a clearing
 with a"+CHR$ 13+"filled up pond
 in it."
512Ø DATA "You are on a grassy h
ill."
5125 DATA "You are at a T-juncti
on in a"+CHR$ 13+"path."
513Ø DATA "This clearing contain
s a large"+CHR$ 13+"pillar. It h
olds up the sky!"
5135 DATA "Fool!!! You have walk
ed into a"+CHR$ 13+"DEATH-TRAP!!
!"
514Ø DATA "In this clearing, the
re is a"+CHR$ 13+"bent tree."
5145 DATA "You are in a clearing
 with a"+CHR$ 13+"ruined house i
n it."
515Ø DATA "You are in a stinking
 swamp."
5155 DATA "You are in a ditch, d
ug into theground."
516Ø DATA "You are on a path."
5165 DATA "In this clearing ther
e is a deadtree."
517Ø DATA "You are walking along
 on a riverbank."
5175 DATA "You are on a bridge,
crossing"+CHR$ 13+"the river."
518Ø DATA "In this clearing ther
e is a treetrunk."
5185 DATA "You are walking by a
dark, oily lake."
519Ø DATA "In this clearing ther
e is a hut."
5195 DATA "You are in an enclose
d field."+CHR$ 13+"There is a co
w here."
52ØØ DATA "You are in a picnic p
lace."+CHR$ 13+"There are tables
 and benches."
52Ø5 DATA "You are in a clearing
."+CHR$ 13+"You see Pharto's roc
ket. It is"+CHR$ 13+"empty."
521Ø DATA "You are in a wood."+C
HR$ 13+"Paths are everywhere."
5215 DATA "You are at a high alt
ar."+CHR$ 13+"There is a temple
to the East."
522Ø DATA "You are at an temple.
"+CHR$ 13+"There is an altar to
the West."
5225 DATA "You are in a garden s
urrounding a small house."
523Ø DATA "You are in a woody cl
earing."+CHR$ 13+"There are tree
s everywhere."
5235 DATA "You're in a clearing
with 2"+CHR$ 13+"trees. There's
a river to the"+CHR$ 13+"West."
524Ø DATA "You are in a small ho
use."+CHR$ 13+"You see Toronto's
 ship."+CHR$ 13+"It is empty."
5245 DATA "You are in a clearing
 with threetrees in it."
525Ø DATA "You are in a clearing
 with a"+CHR$ 13+"hollow log in
it."
5255 DATA "You are in an open pa
sture."
526Ø DATA "You are in a farmyard
. No animalor bird stirs."
5265 DATA "You are at a farmhous
e. It is"+CHR$ 13+"locked and ba
rred."
527Ø DATA "In this clearing, the
re is"+CHR$ 13+"Gasher's rocket.
"
6ØØØ REM h$
6Ø1Ø DATA "human","android","Kor
zaxian","Noxian","Treen","Subter
anean"
7ØØØ REM o$
7Ø1Ø DATA "A ring for a King","A
 gold key for a gold lock","The
map of the castle","This boat ha
s a large hole in it","To light
the way","Recovory potion","Extr
emely rusty"
```



```
   5 PRINT "Stop The Tape!!!!!":
 PAUSE Ø: CLS
  1Ø FOR a=21 TO 1 STEP -1: FOR
c=1 TO 1Ø: NEXT c: PRINT AT Ø,Ø;
 " The Pluto Adventure."(a TO )
  2Ø NEXT a
  3Ø FOR a=21 TO 1 STEP -1: FOR
c=1 TO 1Ø: NEXT c: PRINT AT 2,a-
```

```
1;" By David Peter Knight."(a TO
 ): NEXT a
  40 PRINT AT 0,1; OVER 1;"_____
_______________"
  50 PRINT AT 2,1; OVER 1;"_ ___
_____________ ___"
  60 PRINT AT 5,0;"Do you want i
nstructions?": PAUSE 0
  70 IF INKEY$="n" THEN  GO TO 9
990
  80 IF INKEY$<>"y" THEN  GO TO
60
  90 CLS : PRINT AT 0,4;"Pluto A
dventure";AT 0,4; OVER 1;"_______
________"
 100 PRINT AT 3,0;"  The year i
s 2066, a          settlement ha
s been discovered  below the sur
face. The Mekon is known to have
 personally visitedthe settlemen
t in his personal  craft. His cr
aft is the latest  in spaceship
design. You as the scientist fro
m Earth, cannot     pilot any spa
ceships, except theMekon's delux
e ship. Your        mission is to
 transport to Plutoand come back
 in the Mekon's     ship, leaving
 the Mekon,         stranded in t
he planet."
 110 PRINT '"   In the adventure
, there are  other characters wh
ich are      either friendly or
unfriendly toyou. If you attack
someone, he  will become unfrien
dly. If he isfriendly, he will g
ive you a    clue. An unfriendly
 person will attack you on sight
."
 120 PAUSE 0: CLS : PRINT AT 0,0
;"Vocabulary";AT 0,0; OVER 1;"__
_________"
 130 PRINT '"NORTH, SOUTH, EAST
and WEST."
 140 PRINT '"VOCABULARY      ...P
rint up the                     v
ocabulary."
 150 PRINT '"TAKE, DROP      ...T
ake or drop                     a
n object."
 160 PRINT '"QUIT            ...Q
uit program."
 170 PRINT '"LOAD, SAVE      ...L
oad or save                     p
resent game."
 180 PRINT '"EXAMINE         ...E
xamine object"
 190 PRINT '"ATTACK          ...A
ttack someone"
 200 PRINT '"DRINK           ...D
rink potion"
 210 PRINT '"LOOK            ...D
escribe place"
 220 PRINT #0;"Press a key.": PA
USE 0: CLS
9990 LOAD "Pluto"
```



Program 3

```
  10 LET s$="Data     ": GO SUB
6570: LET ferdo=1000: LET bill=2
000: LET bill1=0: LET gasher=300
0: LET gasher1=0: LET toronto=40
00: LET toronto1=0: LET pharto=5
000: LET pharto1=0: LET mekon=60
00: LET mekon1=0: LET ilko=7000:
 LET ilko1=0
  20 POKE 23658,8
  25 GO SUB mekon: GO SUB ilko:
GO SUB pharto: GO SUB toronto: G
O SUB gasher: GO SUB bill: GO SU
B ferdo
  30 RANDOMIZE
 400 POKE 23692,255
 490 IF c(1,3)<=0 THEN  GO TO 99
90
 491 GO TO 20
 500 REM VOCABULARY
 510 PRINT '''"Vocabulary"''
 520 PRINT '"NORTH, SOUTH, EAST
and WEST"
 530 PRINT '"SAVE, LOAD ...Save
or load the  positions."
 540 PRINT '"TAKE, DROP ...Take
or Drop an     object."
 550 PRINT '"EXAMINE     ...Exami
ne an object."
 560 PRINT '"VOCABULARY ...Print
 this page."
 570 PRINT '"ATTACK      ...Attac
k a character"
 580 PRINT '"DRINK       ...Drink
```

```
."
 590 PRINT '"QUIT        ...Quit
game."
 600 PRINT '"INVENTORY   ...Print
up what you are carrying."
 610 PRINT '"DATA        ...Gives
data on a  character."
 740 PRINT '''"Press a Key.": PAU
SE 0
 749 RETURN
 750 REM EXAMINE
 760 FOR X=1 TO LEN Z$: IF Z$(X)
=" " THEN  GO TO 800
 770 NEXT X
 780 PRINT "Examine WHAT???"
 790 RETURN
 800 LET z$=z$(x+1 TO )
 810 IF z$(1)="R" AND o(1,4)=0 T
HEN  PRINT o$(1)'': RETURN
 820 IF z$(1)="K" AND o(2,4)=0 T
HEN  PRINT o$(2)'': RETURN
 830 IF z$(1 TO 2)="SC" AND o(3,
4)=0 THEN  PRINT o$(3)'': GO TO
3750
 840 IF z$(1)="B" AND o(4,4)=0 T
HEN  PRINT o$(4)'': RETURN
 850 IF z$(1)="T" AND o(5,4)=0 T
HEN  PRINT o$(5)'': RETURN
 860 IF z$(1)="P" AND o(6,4)=0 T
HEN  PRINT o$(6)'': RETURN
 870 IF z$(1 TO 2)="SW" AND o(7,
4)=0 THEN  PRINT o$(7)'': RETURN

 880 PRINT '''"Sorry, can't exami
ne ";z$
 890 PRINT
 999 RETURN
1000 REM Ferdo
1010 LET q=c(1,5): LET p=c(1,6):
 PRINT d$(a(q,p,1))'"You see:";:
 LET yy=0
1020 FOR x=1 TO 7: IF o(x,4)=q A
ND o(x,5)=p THEN  PRINT TAB 10;c
$(1,o(x,1));: LET yy=yy+1
1030 NEXT x
1040 FOR x=2 TO 7: IF c(x,5)=q A
ND c(x,6)=p THEN  PRINT TAB 10;c
$(2,c(x,1));: LET yy=yy+1
1050 NEXT x: IF yy=0 THEN  FOR r
=1 TO 8: PRINT CHR$ 8;: NEXT r:
PRINT "        "
1051 PRINT '"Exits:"
1052 IF a(q,p,2)=1 THEN  PRINT T
AB 6;"North"
1053 IF a(q,p,3)=1 THEN  PRINT T
AB 6;"South"
1054 IF a(q,p,4)=1 THEN  PRINT T
AB 6;"East"
1055 IF a(q,p,5)=1 THEN  PRINT T
AB 6;"West"
1056 IF q=6 AND p=5 THEN  GO TO
9990
1060 IF gasher1=1 THEN  LET gash
er1=0: GO SUB 8000
1070 IF mekon1=1 THEN  LET mekon
1=0: GO SUB 9000
1076 IF q=7 AND p=1 THEN  PRINT
"You have Finished!!!"'"Now you
can go off in the    Mekon's
ship!!!": GO TO 9991
1080 IF ilko1=1 THEN  LET ilko1=
0: GO SUB 8500
1090 IF bill1=1 THEN  LET bill1=
0: GO SUB 9250
1100 IF toronto1=1 THEN  LET tor
onto1=0: GO SUB 9500
1110 IF pharto1=1 THEN  LET phar
to1=0: GO SUB 9750
1210 PRINT '"What Now? ";
1220 INPUT LINE z$
1230 PRINT z$''
1235 LET z$=z$+"
          "
1240 IF z$(1)="N" AND a(q,p,2)=1
 THEN  LET p=p-1: PRINT '''"You g
o North"''
1250 IF z$(1)="S" AND a(q,p,3)=1
 THEN  LET p=p+1: PRINT '''"You g
o South"''
1260 IF z$(1 TO 2)="EA" AND a(q,
p,4)=1 THEN  LET q=q+1: PRINT ''
"You go East"''
1270 IF z$(1)="W" AND a(q,p,5)=1
 THEN  LET q=q-1: PRINT '''"You g
o West"''
1280 IF z$(1)="T" THEN  GO SUB 1
500
1290 IF z$(1)="V" THEN  GO SUB 5
00
1300 IF z$(1 TO 3)="DRO" THEN  G
O SUB 1750
1310 IF z$(2)="X" THEN  GO SUB 7
50
1320 IF z$(1)="K" OR z$(1)="A" T
HEN  GO SUB 2500
1330 IF z$(1 TO 3)="DRI" THEN  G
O SUB 3500
1340 IF o(1,4)=5 AND o(1,5)=4 TH
EN  PRINT "As you drop the ring,
 the statuecomes to life."'"It i
s the King. He summons his  peop
le."'"As a reward, he gives you
the   Mekon's ship, which was in
 his  his room."'"You can escape
```

```
!!!": GO TO 9991
1350 IF z$="LOOK" THEN  GO SUB 4
500
1360 IF q=4 AND p=1 AND o(2,4)<>
0 THEN  PRINT '"But you bump int
o the door,     which is locked.
 You need the    key."': LET p=2
1370 IF z$(1 TO 3)="INV" THEN  G
O SUB 7500
1380 IF z$="QUIT" THEN  GO TO 99
91
1390 IF z$="SAVE" THEN  GO SUB 5
500
1400 IF z$="LOAD" THEN  GO SUB 6
500
1410 IF z$(1 TO 4)="DATA" THEN
GO SUB 2750
1490 LET c(1,5)=q: LET c(1,6)=p
1499 RETURN
1500 REM TAKE
1510 FOR X=1 TO LEN Z$: IF Z$(X)
=" " THEN  GO TO 1600
1520 NEXT x
1530 PRINT '"Can't take nothing!
"'
1599 RETURN
1600 LET Z$=Z$(X+1 TO )
1610 IF z$(1)="R" AND o(1,4)=q A
ND o(1,5)=p THEN  LET o(1,4)=0:
LET o(1,5)=0: PRINT '"Got the Ri
ng"': RETURN
1620 IF z$(1)="K" AND o(2,4)=q A
ND o(2,5)=p THEN  LET o(2,4)=0:
LET o(2,5)=0: PRINT '"Got the Ke
y"': RETURN
1630 IF z$(1 TO 2)="SC" AND o(3,
4)=q AND o(3,5)=p THEN  LET o(3,
4)=0: LET o(3,5)=0: PRINT '"Got
the Scroll"': RETURN
1640 IF z$(1)="B" AND o(4,4)=q A
ND o(4,5)=p THEN  LET o(4,4)=0:
LET o(4,5)=0: PRINT '"Got the Bo
at"': RETURN
1650 IF z$(1)="T" AND o(5,4)=q A
ND o(5,5)=p THEN  LET o(5,4)=0:
LET o(5,5)=0: PRINT '"Got the To
rch"': RETURN
1660 IF z$(1)="P" AND o(6,4)=q A
ND o(6,5)=p THEN  LET o(6,4)=0:
LET o(6,5)=0: PRINT '"Got the Po
tion"': RETURN
1670 IF z$(1 TO 2)="SW" AND o(7,
4)=q AND o(7,5)=p THEN  LET c(1,
4)=c(1,4)+2: LET o(7,4)=0: LET o
(7,5)=0: PRINT '"Got the Sword"'
: RETURN
1680 PRINT '"Sorry can't take th
at."'
1749 RETURN
1750 REM DROP
1760 FOR x=1 TO LEN z$: IF z$(x)
=" " THEN  GO TO 1800
1770 NEXT x
1780 PRINT "Drop What???"
1790 RETURN
1800 LET z$=z$(x+1 TO )
1810 IF z$(1)="R" AND o(1,4)=0 T
HEN  LET o(1,4)=q: LET o(1,5)=p:
 RETURN
1830 IF z$(1 TO 2)="SC" AND o(3,
4)=0 THEN  LET o(3,4)=q: LET o(3
,5)=p: RETURN
1840 IF z$(1)="B" AND o(4,4)=0 T
HEN  LET o(4,4)=q: LET o(4,5)=p:
 RETURN
1850 IF z$(1)="T" AND o(5,4)=0 T
HEN  LET o(5,4)=q: LET o(5,5)=p:
 RETURN
1860 IF z$(1)="P" AND o(6,4)=0 T
HEN  LET o(6,4)=q: LET o(6,5)=p:
 RETURN
1870 IF z$(1 TO 2)="SW" AND o(7,
4)=0 THEN  LET c(1,4)=c(1,4)-2:
LET o(7,4)=q: LET o(7,5)=p: RETU
RN
1999 RETURN
2000 REM Bill
2010 LET Q=C(2,5): LET P=C(2,6):
 LET X=INT (RND*4)+2
2020 IF A(Q,P,X)=0 THEN  RETURN
2030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(2,5)=Q: LET
 C(2,6)=P
2040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET BILL1=1
2499 RETURN
2500 REM ATTACK
2510 FOR x=1 TO LEN z$: IF z$(x)
=" " THEN  GO TO 2550
2520 NEXT x
2540 RETURN
2550 LET z$=z$(x+1 TO )
2560 LET k=0
2570 IF z$(1)="B" THEN  LET k=2
2580 IF z$(1)="G" THEN  LET k=3
2590 IF z$(1)="T" THEN  LET k=4
2600 IF z$(1)="P" THEN  LET k=5
2610 IF z$(1)="M" THEN  LET k=6
2620 IF z$(1)="I" THEN  LET k=7
2630 IF k=0 THEN  PRINT ''"I do
not understand!!"'': RETURN
2635 IF C(K,5)<>C(1,5) OR C(K,6)
<>C(1,6) THEN  PRINT C$(2,K);" i
s not here!": RETURN
```

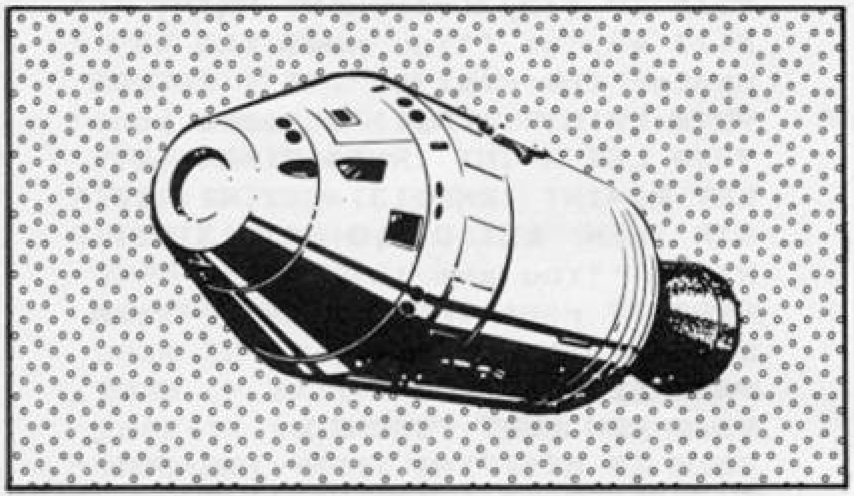
```
2640 PRINT '''You attack ";c$(2,
k);"."
2650 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(k,4): IF R
1>R THEN  PRINT '''He defends yo
ur blow!"
2660 IF r>r1 THEN  PRINT '''You
injure him."'': LET c(k,3)=c(k,3
)-3: IF c(k,3)<0 THEN  PRINT ''c
$(2,k);" is dead.": LET c(k,5)=6
: LET c(k,6)=7
2670 LET c(k,7)=-1
2749 RETURN
2750 FOR x=1 TO LEN z$: IF z$(x)
=" " THEN  GO TO 2800
2760 NEXT x
2790 RETURN
2800 LET z$=z$(x+1 TO )
2805 LET x=0
2810 IF z$(1)="B" THEN  LET x=2
2820 IF z$(1)="G" THEN  LET x=3
2830 IF z$(1)="T" THEN  LET x=4
2840 IF z$(1)="P" THEN  LET x=5
2850 IF z$(1)="M" THEN  LET x=6
2860 IF z$(1)="I" THEN  LET x=7
2870 IF x=0 THEN  PRINT '''No da
ta available."'': RETURN
2880 PRINT 'c$(2,x)''h$(x-1)''
2890 RETURN
2999 RETURN
3000 REM Gasher
3010 LET Q=C(3,5): LET P=C(3,6):
 LET X=INT (RND*4)+2
3020 IF A(Q,P,X)=0 THEN  RETURN
3030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(3,5)=Q: LET
 C(3,6)=P
3040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET GASHER1=1
3499 RETURN
3500 REM DRINK
3510 FOR x=1 TO LEN z$: IF z$(x)
=" " THEN  GO TO 3600
3520 NEXT x
3590 RETURN
3600 IF z$(x+1)<>"P" OR o(6,4)<>
0 THEN  RETURN
3610 LET c(1,3)=c(1,3)+5: LET c(
1,4)=c(1,4)+5
3620 RETURN
3749 RETURN
3750 FOR a=1 TO 10: PRINT AT a,0
;"
  ": NEXT a
3770 PLOT 61,96: DRAW 3,0: DRAW
0,32: DRAW 32,0: DRAW 0,-32: DRA
W 32,0: DRAW 0,32: DRAW 32,0: DR
AW 0,32: DRAW -160,0: DRAW 0,-32
: DRAW 32,0: DRAW 0,-32: DRAW 3,
0
3780 PLOT 32,160: DRAW 0,-6: PLO
T 32,134: DRAW 0,-6: IF O(2,4)<>
0 THEN  DRAW 32,0
3790 PLOT 64,128: DRAW 0,6: PLOT
 64,160: DRAW 0,-6: PLOT 96,160:
 DRAW 0,-6: PLOT 96,134: DRAW 0,
-6: DRAW 6,0: PLOT 122,128: DRAW
 6,0: DRAW 0,6: PLOT 128,160: DR
AW 0,-6
3800 PRINT AT 2,0; OVER 1;"Din H
allSta CorrKing"
3810 PRINT AT 3,0; OVER 1;"Room
  irs idorRoom"
3820 PRINT AT 6,4; OVER 1;"Door
  Bath"
3830 PRINT AT 7,12; OVER 1;"Room
"
3840 PRINT AT 21,31;"  "
3990 PAUSE 0
3999 RETURN
4000 REM Toronto
4010 LET Q=C(4,5): LET P=C(4,6):
 LET X=INT (RND*4)+2
4020 IF A(Q,P,X)=0 THEN  RETURN
4030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(4,5)=Q: LET
 C(4,6)=P
4040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET toronto1=1
4499 RETURN
4500 REM LOOK
4510 RETURN
5000 REM Pharto
5010 LET Q=C(5,5): LET P=C(5,6):
 LET X=INT (RND*4)+2
5020 IF A(Q,P,X)=0 THEN  RETURN
5030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(5,5)=Q: LET
 C(5,6)=P
5040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET pharto1=1
5499 RETURN
5500 REM SAVE
5510 LET S$="         ": INPUT "
Filename to SAVE?";S$(1 TO 9)
5520 SAVE s$ DATA a(): SAVE s$+"
1" DATA o()
5530 SAVE s$+"2" DATA c(): SAVE
s$+"3" DATA c$()
5540 SAVE s$+"4" DATA d$(): SAVE
 s$+"5" DATA h$()
5550 SAVE s$+"6" DATA o$()
5560 PRINT '''Data SAVEd ";
5999 RETURN
```

```
6000 REM Mekon
6010 LET Q=C(6,5): LET P=C(6,6):
 LET X=INT (RND*4)+2
6020 IF A(Q,P,X)=0 THEN  RETURN
6030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(6,5)=Q: LET
 C(6,6)=P
6040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET mekon1=1
6499 RETURN
6500 REM LOAD
6510 LET s$="          ": INPUT "
Filename to LOAD? ";s$(1 TO 9)
6570 LOAD s$ DATA a(): LOAD s$+"
1" DATA o()
6580 LOAD s$+"2" DATA c(): LOAD
s$+"3" DATA c$()
6590 LOAD s$+"4" DATA d$(): LOAD
 s$+"5" DATA h$()
6600 LOAD s$+"6" DATA o$()
6999 RETURN
7000 REM Ilko
7010 LET Q=C(7,5): LET P=C(7,6):
 LET X=INT (RND*4)+2
7020 IF A(Q,P,X)=0 THEN  RETURN
7030 LET Q=Q+(X=4)-(X=5): LET P=
P+(X=3)-(X=2): LET C(7,5)=Q: LET
 C(7,6)=P
7040 IF Q=C(1,5) AND P=C(1,6) TH
EN  LET ilko1=1
7499 RETURN
7500 REM INVENTORY
7510 PRINT '"You are carrying :
"
7520 FOR X=1 TO 7: IF O(X,4)=0 T
HEN  PRINT C$(1,X)
7530 NEXT X
7999 RETURN
8000 REM MEET GASHER
8010 PRINT '"Gasher attacks you
."
8020 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(3,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
8030 IF r>r1 THEN  PRINT '"You
defend his blow."
8040 RETURN
8500 REM MEET ILKO
8510 PRINT '"Ilko attacks you."
8520 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(7,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
8530 IF r>r1 THEN  PRINT '"You
defend his blow."
8540 RETURN
9000 REM MEET MEKON
9010 PRINT '"The Mekon attacks
you."
9020 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(6,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
9030 IF r>r1 THEN  PRINT '"You
defend his blow."
9040 RETURN
9250 REM MEET BILL
9260 IF c(2,7)=1 THEN  PRINT '"
Bill say's: ""You need the key t
oenter the castle"".""': RETURN
9270 PRINT '"Bill attacks you."
9280 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(2,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
9290 IF r>r1 THEN  PRINT '"You
defend his blow."
9300 RETURN
9500 REM MEET TORONTO
9510 IF c(4,7)=1 THEN  PRINT '"
Toronto say's: ""Drop the ring a
tthe King's feet"".""': RETURN
9520 PRINT '"Toronto attacks yo
u."
9530 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(4,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
9540 IF r>r1 THEN  PRINT '"You
defend his blow."
9550 RETURN
9750 REM MEET PHARTO
9760 IF c(5,7)=1 THEN  PRINT '"
Pharto say's: ""Go to the King's
 bedroom to escape"".""': RETURN

9770 PRINT '"Pharto attacks you
."
9780 LET R=INT (RND*12)+C(1,4):
LET R1=INT (RND*13)+C(5,4): IF R
1>R THEN  LET C(1,3)=C(1,3)-5: P
RINT '"You are injured"
9790 IF r>r1 THEN  PRINT '"You
defend his blow."
9800 RETURN
9990 PRINT '"You are DEAD!!!"
9991 FOR A=1 TO 100: PAUSE 0: PR
INT '''''''''''''''''''''''': IN
PUT "Another go?";a$
9992 IF a$="Y" THEN  GO TO 10
9993 INPUT "Sure?";a$: IF a$="N"
 THEN  GO TO 10
9994 STOP
```

QL Characteristics

David Nowotnik, fresh from his machine code series, looks at the QL and discovers an interesting UDG ability.



128	ä	C ESC	149	î	C 6	170	Æ	CS k
129	ã	CS 1	150	ó	C 7	171	Œ	CS l
130	å	CS '	151	ò	C 8	172	α	CS m
131	é	CS 3	152	ô	C 9	173	δ	CS n
132	ö	CS 4	153	ú	CS ;	174	θ	CS o
133	õ	CS 5	154	ù	C ;	175	λ	CS p
134	ø	CS 7	155	û	CS ,	176	μ	CS q
135	ü	C '	156	β	C =	177	π	CS r
136	ç	CS 9	157	¢	CS .	178	ϕ	CS s
137	ñ	CS 0	158	¥	CS /	179	¡	CS t
138	æ	CS 8	159	`	CS 2	180	¿	CS u
139	œ	CS =	160	Ä	CS a	181	ş	CS v
140	á	C ,	161	Ã	CS b	182	§	CS w
141	à	C -	162	Å	CS c	183	¤	CS x
142	â	C /	163	É	CS d	184	«	CS y
143	ë	C 0	164	Ö	CS e	185	»	CS z
144	è	C 1	165	Õ	CS f	186	°	C [
145	ê	C 2	166	Ø	CS g	187	÷	C \
146	ï	C 3	167	Ü	CS h	188	←	C]
147	í	C 4	168	Ç	CS i	189	→	CS 6
148	ì	C 5	169	Ñ	CS j	190	↑	CS -

Table 1. The full QL character set.

In addition to a fairly standard ASCII character set, with code numbers between 32 and 127, the QL offers an extra character set, with code numbers between 128 and 190. The QL handbook, as with many other things, doesn't give away any details about these extra characters. You may have noticed some of them mysteriously appearing on the screen when by accident the CTRL key was pressed with another key. A full list of these extra characters appears in Table 1.

Table 1 indicates the character code, the character itself, and the combination of keys you need to press to get the character on the screen. In the table, 'C' stands for CTRL and 'S' for shift, so for some of these extra characters you'll have to press three keys at once. For example, the 'right' arrow (character code 189) is obtained by simultaneously pressing SHIFT, CTRL, and '6'.

Apart from the arrows and chevrons, these extra characters are of little use unless you have an interest in foreign languages. Far more useful would be the capability to produce user-defined graphic characters (UDGs). Again, it isn't obvious from the Sinclair documentation how this might be done, but the QL does allow UDGs; the trick is in understanding how characters are produced.

Every channel to an exterior 'device', including the screen, has a block of RAM allocated to it which contains important information for the management of that channel. The start of this information is address 167424, decimal. For every channel opened to the screen (including channels 0,1, and 2) there are two addresses stored which point to the start of the two blocks of character bit patterns in ROM. One block holds the bit patterns for characters 32 to 127, and the other for characters 128 to 190. These bit patterns define how a character will look when it appears on the screen.

Figure 1. UDG Designer program.

```
100 REMark          UDG Designer
110 REMark      by David Nowotnik
120 REMark           June, 1985
130 REMark
500 init: instructions: ch=char
510 grid: add_cursor
520 REPeat qq
530   selection
540   CLS#2: INK 0
550   FOR i=1 TO 9
560   x=0
570     FOR j=1 TO 6
580       IF a$(i,j)="1" THEN x=x+2^(7-j)
590     NEXT j
600   POKE addr+10+9*(ch-128)+i,x
610   AT i,17: PRINT x
620   NEXT i
630 AT #2,1,12: PRINT #2,"UDG DESIGNER"
640 PRINT #2,\"  Character ";ch;" is:-"
650 PRINT #2,\ TO 6, CHR$(ch)
660 PRINT #2,\: CSIZE #2,1,1: PRINT #2, TO 4, CHR$(ch)
670 PRINT #2,\: CSIZE #2,2,1: PRINT #2, TO 2, CHR$(ch)
680 CSIZE #2,0,0
690 PRINT #2,\\" Is this OK? (y/n)"
700 REPeat yes_or_no
710   x$=INKEY$ (-1)
720   IF x$="y" OR x$="Y" OR x$="n" OR x$="N" THEN EXIT yes_
or_no
730 END REPeat yes_or_no
740 IF x$="Y" OR x$="y" THEN EXIT qq
750 instructions: FOR t=1 TO 2
760 IF a$(row,col)="0" THEN
770           INK 7:square: a$(row,col)="1": INK 2: add_cursor
780         ELSE
790         INK 2: square: a$(row,col)="0": INK 7: add_cursor
800         END IF
810 END FOR t
850 END REPeat qq
860 PRINT #2,\\"        Any more (y/n)?"
870 x$= INKEY$(-1)
880 IF x$="y" OR x$="Y" THEN RUN
900 STOP
1000 REMark ****************************
1010 DEFine PROCedure grid
```

```
1020 REMark ***************************
1030 LOCal i,j: CLS: SCALE 100,0,0
1040 FOR i=20 TO 65 STEP 9
1050 LINE i,1 TO i,91
1060 END FOR i
1070 FOR j= 1 TO 91 STEP 10
1080 LINE 20,j TO 65,j
1090 END FOR j
1100 END DEFine
1500 REMark ***************************
1510 DEFine PROCedure init
1520 REMark ***************************
1530 col=1: row=1
1540 DIM a$(9,6)
1550 FOR i=1 TO 9
1560   a$(i)="000000"
1570 END FOR i
1580 CSIZE 2,1
1590 INK 7: PAPER 2: CLS
1600 IF PEEK_L (167726)> 50000 THEN addr=PEEK_L (167726): RE
Turn
1610 addr=RESPR (875)
1620 FOR i=0 TO 875 STEP 4
1630   POKE_L addr+i, PEEK_L (PEEK_L(167726)+i)
1640 NEXT i
1650 POKE_L 167982,addr
1660 END DEFine
2000 REMark ***************************
2010 DEFine PROCedure instructions
2020 REMark ***************************
2030 CLS#2: AT #2,1,12: PRINT #2,"UDG DESIGNER"
2040 AT #2,4,2: PRINT #2,"This program allows you to design
your  own user defined graphics. The program    stores the n
ew UDGs in reserved space."
2050 PRINT #2,\"  Move the cursor '*' around with the      cu
rsor control keys. Press 'space' if you want to alter the pi
xel at the cursor     position."
2060 PRINT #2,\"  Press 's' when you want to stop and      ex
amine the udg at normal scale."
2200 END DEFine
2500 REMark ***************************
2510 DEFine FuNction keypress
2530 LOCal z$
2540 REPeat loop
2550   z$=INKEY$(-1)
2560   IF z$=" " THEN RETurn 1
2570   IF z$=CHR$(192) THEN RETurn 3: REMark left
2580   IF z$=CHR$(200) THEN RETurn 4: REMark right
2590   IF z$=CHR$(208) THEN RETurn 5: REMark up
2600   IF z$=CHR$(216) THEN RETurn 6: REMark down
2610   IF z$="s" OR z$="S" THEN RETurn 2
2620 END REPeat loop
2650 END DEFine
3000 REMark ***************************
3010 DEFine PROCedure selection
3020 REMark ***************************
3030 LOCal z
3040 REPeat options
3050   z=keypress
3060   SELect ON z
3070   =1: IF a$(row,col)="0" THEN
3080         INK 7:square: a$(row,col)="1": INK 2: add_curso
r
3090       ELSE
3100       INK 2: square: a$(row,col)="0": INK 7: add_cursor
3110       END IF
3120   =2: RETurn
3130   =3: IF col>1 THEN
3140         add_cursor: col=col-1: add_cursor
3150       END IF
3160   =4: IF col<5 THEN
3170         add_cursor: col=col+1: add_cursor
3180       END IF
3190   =5: IF row>1 THEN
3200         add_cursor: row=row-1: add_cursor
3205       END IF
3210   =6: IF row<9 THEN
3220         add_cursor: row=row+1: add_cursor
3230       END IF
3240    END SELect
3250 END REPeat options
3260 END DEFine
3500 REMark ***************************
3510 DEFine FuNction char
3520 REMark ***************************
3530 LOCal z
3540 REPeat get_char
3550   CLS#0: INPUT #0,"  Enter a character code (128-190) "
; z
3560   IF z=INT (z) AND z<191 AND z>127 THEN EXIT get_char
3570 END REPeat get_char
3580 PRINT #2,\"   Character code = ";z: CLS #0
3590 RETurn z
3600 END DEFine
4000 REMark ***************************
4010 DEFine PROCedure add_cursor
4020 REMark ***************************
4030 OVER -1
4040 AT row,3+2*col: PRINT "*"
4050 OVER 0
4060 END DEFine
5000 REMark ***************************
5010 DEFine PROCedure square
5020 REMark ***************************
5030 LOCal k: SCALE 300,0,0
5040 FOR k=3*(12+9*(col)) TO 3*(19+9*(col))
5050   LINE k,3*(2+10*(9-row)) TO k,3*(10+10*(9-row))
5060 END FOR k
5070 END DEFine
10000 REMark ***************************
10010 DEFine PROCedure screen
10020 REMark ***************************
10030 MODE 4
10040 WINDOW #1,256,205,256,0
10050 WINDOW #2,256,205,0,0
10060 BORDER #1,2,0: BORDER #2,2,0
10070 INK 7: PAPER 2: INK #2,2: PAPER #2,7
10080 CLS: CLS #2
10090 END DEFine
```

These stored addresses (vectors) in the information block on each channel always appear at the same position relative to the start of the information block. The vector to the standard character set is 42 bytes from the start, and the 'extra' character set is 46 bytes from the start. So, if you can find the start of the information block, the vector address will be easily located. And once the vector address is found, you can POKE in a new value, to point to your own bit pattern table — in other words, your own UDGs.

Channels

The easiest way to find the start of a particular channel information block is to refer to another information block in RAM, which (in the 'JM' version, at least) starts at 166752. Using PEEK_L (166752) will return the start address of the channel 0 information block; the same PEEK on addresses 166756 and 166760 will give channel 1 and channel 2 start addresses, respectively. As more channels are OPENed, their information block start addresses appear 166764, 166768, etc.

So, the vector to the 'extra' character bit pattern block in channel 2 is ; PEEK_L (166760)+46, and PEEK_L of the resultant address will give the start address of the bit pattern block in ROM (unless you have changed the vector).

The fact that each channel has its own vector to the character bit patterns provides the QL with a very powerful feature. You can redefine characters for one channel only (or more if you want) whilst retaining the original ones for other channels. Or, you could have several blocks of UDGs, a different one for each channel!

UDG design

The simpliest way to design your own UDGs is to have a QL program do it for you. My UDG design program is featured as Listing 1. Type in the listing, and save it. The best way to understand how it works is to see it in operation.

The program is designed to work with the 'monitor' screen layout of the QL. If you have selected the 'TV' option, then the program contains a procedure called 'screen' (which you can type in as a direct command before running the program) to give you the required screen layout. In line 1610 some RAM is reserved, into which the 'extra' character set bit-pattern is copied from ROM (Lines 1620-1640). Line 1650 resets the channel 2 (left-hand window) 'extra' character set vector to the new start address in RAM. So the UDGs will be created for channel 2, and only characters of codes 128 to 190 will be affected, and these will only be changed if they are selected by you when running the UDG design program.

Having selected which character to redesign, a grid of 9 × 5 is drawn in the right hand window which represents an enlarged character. All the QL's characters are made up from a 9 × 5 grid of dots. Move the cursor ("✱") around the grid with the cursor control keys, pressing the space bar to create/erase a dot in any one position of the matrix. When you want to view ▶

Figure 2. Lunar Lander program.

```
10 REMark          Lunar Lander Game
20 REMark          by David Nowotnik
30 REMark            June, 1985
100 initialise
110 udgs: stars: instructions
140 begin: control_board
150 start
160 REPeat main_loop
170   setting=keypress (setting)
180   rocket (setting)
190   calculations
200   readings
210   rocket (setting)
220   IF height<4000 THEN EXIT main_loop
240   shift
250 END REPeat main_loop
260 landing 0,0
270 REPeat end_play
280   setting=keypress(setting)
290   rocket (setting)
300   calculations
310   readings
330   rocket (setting)
340   landing mv,1
350   IF height<=0 THEN EXIT end_play
360 END REPeat end_play
370 AT 10,1: speed=-1*( INT (velocity)+.5): BEEP: INK 2
380 SELect ON speed
390  ON speed=0 TO 15: rocket 0: PRINT " Safe landing"
400  ON speed=16 TO 40: rocket 0
410      AT #3,2,0: PAPER #3,6: CLS #3,2
420      PRINT "You sustained serious            damage"
430  ON speed=REMAINDER
440      BEEP 0,30,0,0,0,0,12
450      AT 5,0
460      FOR i=1 TO 8
470      PAPER 7: CLS 1
480      FOR k=1 TO 100: NEXT k
490      PAPER 0: CLS 1
500      FOR k=1 TO 100: NEXT k
510      END FOR i
520 AT 10,1: PRINT " You killed the crew!"
530 BEEP
550 END SELect
560 CLS #4: AT #5,8,0
570 PRINT #5," Another game?"\\"  (y/n)"
580 REPeat rr
590   z$=INKEY$ (-1)
600   IF z$="Y" OR z$="y" THEN RUN
610   IF z$="n" OR z$="N" THEN EXIT rr
620 END REPeat rr
650 STOP
1000 REMark ******************************
1010 DEFine PROCedure udgs
1020 REMark ******************************
1030 LOCal a,n,byt,udg_start
1040 a= RESPR(0):IF a>261900 THEN a=RESPR (244)
1050 udg_start=261900
1060 POKE_L (PEEK_L (166764)+46), udg_start
1070 RESTORE 1080
1080 FOR n=0 TO 100
1090   READ byt: POKE (udg_start + n), byt
1100 END FOR n
1110 DATA 127,15
1120 DATA 0,0,0,0,0,0,0,0,0
1140 DATA 0,32,60,60,60,32,32,32,32
1150 DATA 56,56,56,56,124,124,124,124,124
1160 DATA 0,0,4,12,28,60,124,124,124
1170 DATA 124,124,124,124,124,124,124,124,124
1180 DATA 0,0,64,96,112,120,124,124,124
1190 DATA 12,12,24,48,96,96,96,96,112
1200 DATA 96,96,48,24,12,12,12,12,28
1210 DATA 124,56,56,16,0,0,0,0,0
1220 DATA 124,124,56,56,56,56,16,0,0
1230 DATA 124,124,56,56,56,56,56,16,0
1240 END DEFine
1500 REMark ******************************
1510 DEFine PROCedure initialise
1520 REMark ******************************
1530 LOCal x
1540 MODE 8: WINDOW #1, 312,205,200,0
1550 OPEN #3, scr: OPEN #4, scr: OPEN #5, scr
1560 WINDOW #3, 40,70,350,20: CSIZE #3,2,0
1570 WINDOW #4, 200,205,0,0: CSIZE #4,0,0
1580 INK 7: PAPER 0: CLS
1590 INK #4, 2: PAPER #4,6: CLS #4
1600 BORDER #4,5,3
1610 WINDOW #5, 180,165,10,40: CSIZE #5,0,0
1620 PAPER #5,6: INK #5,0
1700 END DEFine
2000 REMark ******************************
2010 DEFine PROCedure rocket (x)
2020 REMark ******************************
2030 LOCal i
2040 OVER #3,-1: AT #3,0,0
2050 INK #3,3: PRINT #3," "& CHR$ (128)&" "
2060 INK #3,4: PRINT #3," "& CHR$ (129)&" "
2070 INK #3,7: PRINT #3, CHR$ (130);
2080 INK #3,4: PRINT #3, CHR$ (131);
2090 INK #3,7: PRINT #3, CHR$ (132)
2100 INK #3,6: PRINT #3, CHR$ (133)&" "& CHR$ (134)
2120 INK #3,2: y=3
2130 REPeat flames
2140   IF x=0 THEN EXIT flames
2145   AT #3,y,1
2150   IF x>3 THEN
2160      PRINT #3, CHR$(131)
2170      x=x-4
2180   ELSE
2190      PRINT #3, CHR$ (134+x): x=0
2200   END IF
2210   y=y+1
2220 END REPeat flames
2230 OVER #3,0
2240 END DEFine
2500 REMark ******************************
2510 DEFine PROCedure stars
2520 REMark ******************************
2530 LOCal i
2540 FOR i=1 TO 80
2550   POINT RND (112), RND (100)
2560 NEXT i
2570 END DEFine
2600 REMark ******************************
2610 DEFine PROCedure more_stars
2620 REMark ******************************
2630 LOCal i,x,x1
2650 IF mv<0 THEN
2660   x=0: x1=-mv/2
2670   ELSE
2680   x=100-mv/2: x1=100
2690 END IF
2700 FOR i=0 TO ABS (mv)/5
2710   POINT RND (112), RND (x TO x1)
2720 NEXT i
2730 END DEFine
3000 REMark ******************************
3010 DEFine PROCedure delay (x)
3020 REMark ******************************
3030 LOCal j,k
3040 FOR j=1 TO 100
3050    FOR k=1 TO x
3060    NEXT k
3070 NEXT j
3080 END DEFine
3200 REMark ******************************
3210 DEFine PROCedure start
3220 REMark ******************************
3230 height=55000
3240 velocity=-180
3250 fuel=3000: accel=0
3260 setting=0: dr=1: mv=9
3270 ADATE (-DATE): time=0
3280 POKE_W 163886,0: old_time=0
3290 END DEFine
3400 REMark ******************************
3410 DEFine PROCedure calculations
3420 REMark ******************************
3430 new_time= PEEK_W (163886)
3440 update=(new_time-old_time)/50
3450 old_time=new_time: sign=1
3460 accel=5.4*update*(.2*setting -1): IF accel<0 THEN sign=
-1
3470 change_in_height= update*(velocity+sign*(accel^2))
3480 height=height+change_in_height
3490 velocity=(change_in_height/update)
3500 mv=(change_in_height)/25
3510 fuel=fuel-update*setting*2
3520 IF fuel<=0 THEN fuel=0
3530 IF height<0 THEN height=0
3550 END DEFine
3600 REMark ******************************
3610 DEFine FuNction keypress (x)
3620 REMark ******************************
3630 LOCal z$,z
3640 z$= INKEY$ (0)
3650 IF z$="" OR z$<"0" OR z$>"9" THEN
3660    z=x: ELSE
3665    z=z$: END IF
3670 IF fuel=0 THEN z=0
3680 IF z=0 THEN BEEP: RETurn z
3690 BEEP 0,90/z,0,0,0,0,12
3700 RETurn z
3710 END DEFine
4000 REMark ******************************
4010 DEFine PROCedure control_board
4020 REMark ******************************
4030 title
```

```
4040 AT #4,5,0: PRINT #4,"Height"
4050 PRINT #4,\"Speed"\\"Fuel"\\"Time"\\"Thrust"
4060 END DEFine
4200 REMark ******************************
4210 DEFine PROCedure title
4220 REMark ******************************
4230 CLS #4
4240 INK #4,1: PRINT #4,\" Lunar Landing"
4250 INK #4,3
4260 END DEFine
4400 REMark ******************************
4410 DEFine PROCedure readings
4420 REMark ******************************
4430 AT #4,5,7: PRINT #4,INT (height);"  "
4440 AT #4,7,7: PRINT #4,-INT (velocity);"  "
4450 AT #4,9,7: PRINT #4,INT (fuel);" "
4460 time$=DATE$: AT #4,11,7: PRINT #4, time$ (16 TO 20)
4470 AT #4,13,7: PRINT #4,setting
4480 END DEFine
4600 REMark ******************************
4610 DEFine PROCedure shift
4620 REMark ******************************
4630 IF mv=0 THEN RETurn
4640 SCROLL mv
4650 more_stars
4660 END DEFine
5000 REMark ******************************
5010 DEFine PROCedure landing (x,y)
5020 REMark ******************************
5040 IF y=0 THEN add= (height-3500)/25
5050 add=add+x
5060 IF ABS (add) >= 1 THEN
5070    IF add >0 THEN
5080       PAPER 0: SCROLL INT (add)
5090       more_stars (add): add=add- INT (add)
5100    ELSE
5110       PAPER 6: SCROLL (INT (add) +1)
5120       add=add- INT (add) -1
5130    END IF
5140 END IF
5150 END DEFine
6000 REMark ******************************
6010 DEFine PROCedure instructions
6020 REMark ******************************
6030 title
6040 PRINT #4\\\"  Do you want "\" instructions?"\"     (y/n
)"
6050 REPeat yn
6060   z$= INKEY$ (-1)
6070   IF z$="Y" OR z$="y" THEN EXIT yn
6080   IF z$="N" OR z$="n" THEN RETurn
6090 END REPeat yn
6100 CLS #5
6110 RESTORE 6600
6120 FOR w=1 TO 26
6130   delay (3)
6140   READ z$: PRINT #5,z$\
6150 NEXT w
6160 delay (10)
6500 END DEFine
6600 DATA "The object of"
6610 DATA "this game is"
6620 DATA "to land your"
6630 DATA "lunar lander on"
6640 DATA "to the surface"
6650 DATA "of the moon."
6660 DATA "You must land"
6670 DATA "with a speed of"
6680 DATA "less than 16 ft"
6690 DATA "per second. You"
6710 DATA "start at 55000"
6720 DATA "feet at 180 ft"
6730 DATA "per second, and"
6740 DATA "3000 units of"
6750 DATA "fuel. Use the"
6760 DATA "number keys, 0"
6770 DATA "to 9 to select"
6780 DATA "rocket thrust."
6790 DATA "A setting of 5"
6800 DATA "will just match"
6810 DATA "the pull of"
6820 DATA "gravity."
6830 DATA "Good luck!"
6840 DATA " "
6850 DATA " "
6860 DATA " "
7000 REMark ******************************
7010 DEFine PROCedure begin
7020 REMark ******************************
7040 CLS #5
7050 AT #5,8,0: PRINT #5," Press a key"\\" to start."
7060 z$= INKEY$ (-1)
7070 END DEFine
```

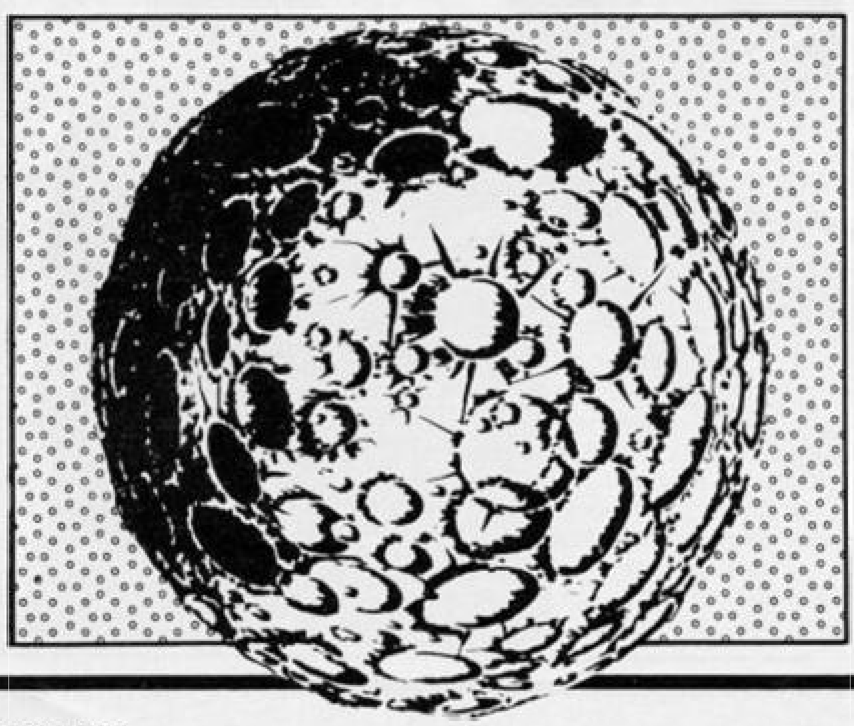
your UDG, press 's'; it will appear in the left hand window at three different character sizes. The nine numbers appearing on the right hand side of the matrix are the numbers POKEd into the bit pattern table (at the appropriate place for that character code, Line 600) to create the new character.

Lunar Lander

Having created one or more UDGs, then these new characters have to be transferred to the application program requiring them. An example of this appears in the program in listing 2. This is a lunar landing program, in wbich the UDGs provide the shape of the landing vehicle. Full game instructions appear with the program, and it's quite fun, as well as being a good example of UDGs!

The UDGs are set up in their own procedure (Line 1000 onwards), in which space is reserved (Line 1040), and the bit pattern start address is defined (Line 1050). Channel 4's 'extra' character set vector is changed to the new start address (1060), and the table contents POKEd into position (1070-1100).

It's worth making some comment about the layout of the character bit pattern table. The first byte of the table contains the character code of the first character in the table *minus 1*. In the 'Lander' program it is 127, as the first character is 128. The second byte contains the number of characters whose bit pattern is defined in the table; fifteen in this example. The next nine bytes make up the bit pattern of the 'default' character — that is, the character which is printed on the screen if you try to print an 'illegal' character. Finally, we get round to the bit patterns of the UDGs, which follow the default character in blocks of nine, in ascending numerical order. The first block of nine defines character 128, the next nine, 129, and so on. The numbers that are POKEd in, in each block of nine, are those which appear in the UDG designer program, and they are POKEd into the bit pattern table in the same order as they appear on the screen in the designer program.



So, you should now have all the information necessary to design and use UDGs in your own programs. Before I close, there are a couple of additional features of 'Lander' (nothing to do with UDGs) which programmers might find of value. The simple statement in Line 3270 sets the internal clock to zero, to make the start of the game. The duration, in minutes and seconds then gets printed onto the screen using a substring (DATE$).

For the calculations involving time, it was necessary to have at least a tenth of a second accuracy; the internal clock was inadequate for this, as it has one second as its smallest unit. The program makes use of the QL's frame counter, which increments every 1/50th of a second. The frame counter is at address 163886, and is a 'word' (two bytes) in length. Having set the frame counter to zero (line 3280), it is then possible to PEEK_W the frame counter to determine the number of 50ths of a second which have elapsed. ■

Peripheral Problems

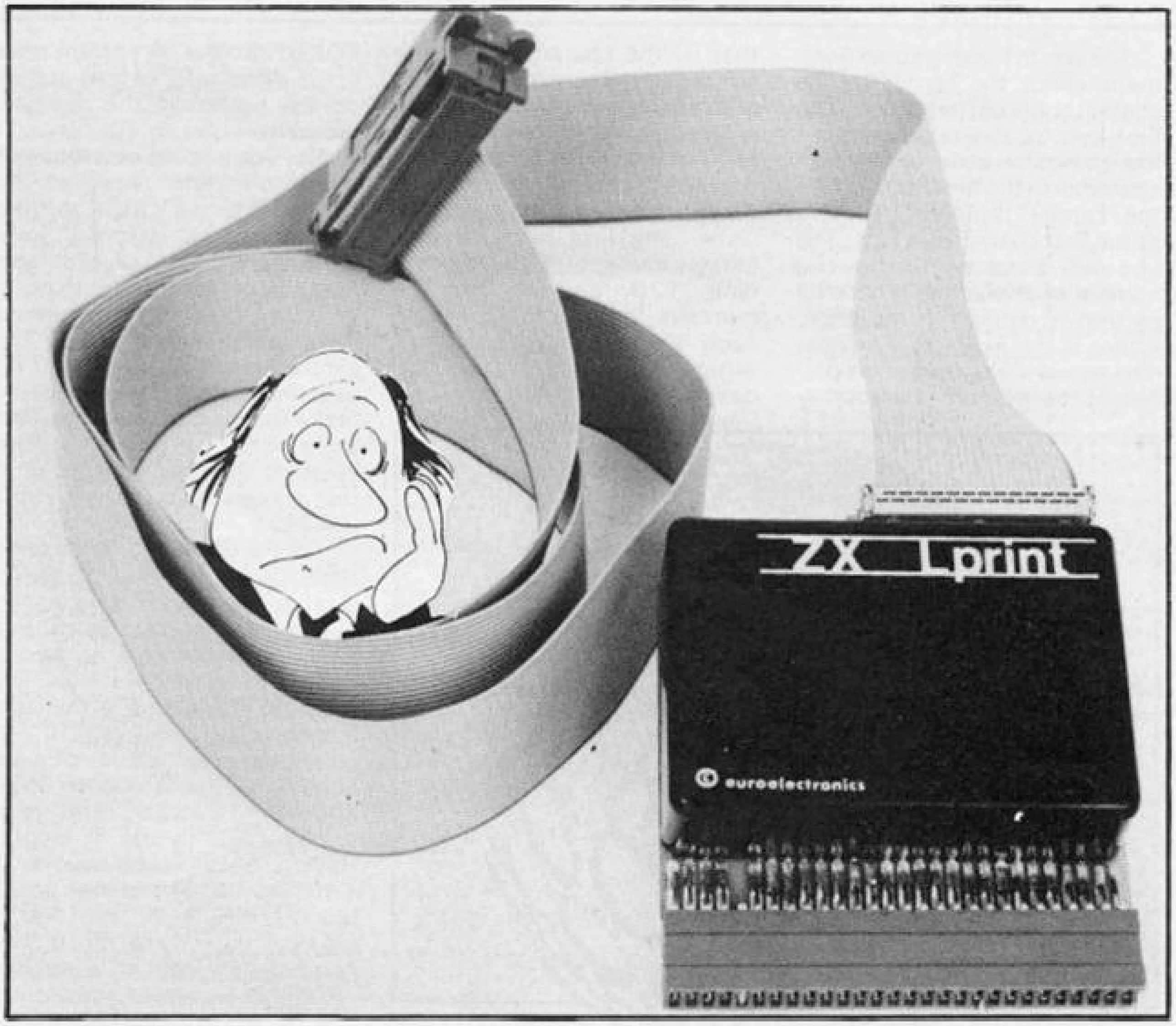
John Ure from Birmingham, has a point to make.

Regular readers of *ZX Computing* may remember my article on the trials and tribulations of attaching a daisywheel printer, interface, and professional keyboard to the ZX Spectrum. At the end of that article I mentioned that I had finally got the system working, apart from a few mysterious "crashes" which I put down to faulty software. As it turned out, the software was not to blame — it was the Spectrum's ULA (Uncommitted Logic Array) which was allergic to the professional keyboard!

The system I was trying to run included a printer interface (ZX LPrint III from Euroelectronics); a daisywheel printer (Smith-Corona TP1); the Fuller FDS keyboard and the Word Processor program, Tasword Two (from Tasman Software).

Since Fuller had gone into liquidation, I was unable to get any help from that source. However, Tasman Software recommended the use of the Transform keyboard in conjunction with a special keyboard buffer which would overcome the problems I was experiencing with the Spectrum's ULA. (I should point out that not all versions of the Spectrum will require this additional "buffering". The only way of finding out if your Spectrum needs a buffer is by attaching the new keyboard and waiting for a program to crash!)

The Transform keyboard and buffer were duly purchased (the first keyboard had to be returned as the GPO seemed to have used it as a football), the Spectrum PCB was taken out of the Fuller case and installed in the Transform case but, how to fit the buffer? Despite much twisting and turning, and many muttered oaths, that buffer had me foxed. A call to Transform, and all was revealed. Two seconds later the buffer was attached and my face had gone a bright red because of the certain knowledge that a five-year-old could have solve the matter. If only firms would send out proper instructions — anything to avoid having to feel a "proper Wally".



Readers will be relieved to know that the system now works perfectly — it's only taken me about four months to get everything "up and running"! That brings me on to another bug-bear of mine. I'm sure I would not have had as many problems if I had been able to deal directly with a specialist computer shop. Why are there so few and why are the so-called "computer departments" in large stores so abysmal? Allow me to explain further. . .

On deciding to purchase a programmable joystick interface (after the excellent review in October's *ZX Computing* — crawl!), I tried several computer outlets in Birmingham without success. A specialist computer store (newly opened) said "Try again next week. We are due to get some in then" (sounds like a TV show!). Unfortunately, I had decided to buy the Fox interface and the store would not be stocking that particular model. I then tried those infamous computer departments. "What's that, then?" "We've got the Sinclair one, will that do?" "Is this what you mean?" (whilst holding a Kempston non-programmable interface) were just some of the more intelligent responses. Once more I would have to purchase by post.

Oh well, perhaps I'll buy some software to cheer myself up, I thought. You've guessed it, no such luck. I couldn't get "Jet Set Willy", "Trashman", "Tornado Low Level" or any of the other dozen or so pieces of software I would have settled for. Perhaps Ray will send me some software to review! Until then I'll continue to buy mail-order despite the occasional hiccup.

(PS — I would have liked to have done a review of a Spectrum monitor interface on offer from a company called "Microdelerium". Unfortunately, after waiting some three months the firm have returned my cheque saying that they are having production problems. They seem to have spelling problems as well!)

Road Race



David Hanson and Geoffrey Bowers, of SMUG — Speyside Microcomputer Users Group — sent us this superior racing game which needs skill and strategy as well as fast reactions.

The Road Race program is somewhat different from others that you may have seen and incorporates several interesting features:

1. The object of the game is to complete five laps in the fastest possible time. A record time is kept.

2. The track covers two screens, has real hairpin bends which involve some careful planning, has PITS for re-fuelling and tyre change, spectators, trees etc, and the screen scrolls downwards.

3. Driving:
You have five possible movements, three of which require two keys to be pressed at the same time.

To negotiate the hairpin bends effectively requires skill.

You decide whether or not to go into the pits and then have the option to take on fuel and/or change tyres.

There are some risky 'stunts' which can be of benefit to an experienced driver, but you can discover these for yourself!

There is a PANIC facility — holding down the 'P' key will temporarily halt you whilst you collect your thoughts!

4. On-screen dials (PRINTed on lines 22/23, which are not affected by the down scroll, but are not COPYed by the printer) show FUEL, TYRE WEAR, LAPS COMPLETED and TIME.

5. Your attempt ends prematurely if you; crash into a tree (asterisk), run over a spectator (inverse zero), run out of fuel, have a burst tyre (because your tyres have worn too much) or run out of space (ie if you get too high up or, more likely, too low down the screen).

Changing the Circuit

If you do not like the circuit as it is, you can easily change it by altering the graphics in the strings in lines 1020 to 1450. It is best to plan your circuit on squared paper first, though.

Notes on some of the Lines

Lines	
7110-7116	Check the fuel input to make sure that a letter key has not been pressed by mistake, since this would BREAK the program with an error report.
5945-5955	PRINTing on lines 22/23 is accessed by POKE 16418,0 before and POKE 16418, 2 afterwards.
9990-9995	If the program is SAVEd by GOTO 9990, the program will auto-RUN on LOADing.

Instructions

The program makes use of two machine code routines
(i) A screen inversion routine.
(ii) A down-scroll routine.

Those readers who are familiar with entering machine code will no doubt have their favourite way of doing this, but for the sake of those who may be doing

this for the first time, a "user-friendly" routine is given in Listing 1.

Enter Listing 1, taking great care that Lines 1 and 2 are correct. These lines will eventually hold the two machine code routines.

Study Listing 2. From left to right, the four columns are: Address — Decimal code — Character — Running total.

The two right-hand columns are for you to check that you have correctly entered the decimal code. The character shows how the machine code will eventually appear in the listing.

The computer will PRINT an address (starting at 16514) and give you a prompt (inverse L) to enter a number. Enter the number in the second column. The computer will automatically PRINT the character in column 3 and the total in column 4, exactly as in Listing 2.

The computer will then wait for you to INPUT a string (inverse L again). Check carefully that the last line on the screen is exactly as shown in Listing 2. If all is well, press Z and NEWLINE. If you have made an error, press any other key and NEWLINE, and you will be able to enter the correct decimal code before the computer goes on to the next address.

Note: when the screen is full, you will get an 'error report 5/9510'. When this happens, press CONT and NEWLINE. Also, there is a jump in address from 16532 to 16539 because the REM in line 2 of the listing occupies addresses 16533-16538 inclusive. If you have made an error, then EDIT and re-RUN this program.

From now on, Line 1 and 2 must be safeguarded, experts will know of a fool-proof method but it is useful for a relative beginner to keep the two machine code routines separate).

You can check the two machine code routines by adding the 5 lines of Listing 3 (taking great care not to eliminate lines 1 and 2!). Then, type RUN and NEWLINE. If all is well, you should get a black screen which scrolls downwards, leaving two black lines at the bottom of the screen. If this does not happen, check where you went wrong!

Now, carefully eliminate lines 10-50 and 9500-9610 (but do NOT eliminate Lines 1 and 2). At this point, SAVE the program (if you keep a copy on a separate cassette, you will be able to use these routines in future, in other programs).

Now enter the lines of Listing 4 (except Lines 1 and 2 which are already there). It is strongly recommended that you SAVE the program every half hour or so in case of a power failure or other problems. SAVEing on alternate sides of the cassette is an additional safeguard.

Finally, SAVE the completed program by GOTO 9990 which will ensure that it auto-RUNs on LOADing.

Cassette Insert

It is relatively easy to design your own cassette insert, particularly if you have a ZX printer which produces screen dumps of the correct size. Why not make use of the one here, or better still design your own?

Variables Used

FT	Fastest time.
T$(44,32)	Track graphics.
N	Number of T$ to be PRINTed.
E	Line number at which T$ is to be PRINTed.
G,F	Line and column number of car.
N1	A value of N used in checking crashes.
Z$	Input of fuel.
LZ	Length of Z$.
Z	Number of character in Z$.
Z	Fuel input.
Y	Fuel during re-fuelling routine.
R	Random time for tyre change.
T	Time during tyre change routine.
R$	Input during end routine.
S$,V$	Title.
LAPS	Number of completed laps.
FUEL	Fuel left.
TW	Tyre wear.
TIME	Time taken so far.

Main Sections of the Program. These are indicated in the listing as REMs in inverse video, but are listed here for convenience.

100	INSTRUCTIONS
1000	TRACK GRAPHICS
1500	PRINT RECORD TIME (If appropriate)
3000	PRINT TRACK AND CAR AT THE START
3200	START VARIABLES
5000	CONTROL CENTRE (The main part of the program, checking to see if keys have been pressed, PRINTing car, updating the variables FUEL, TW, TIME).
5900	PRINT FUEL ETC
6000	CHECK CRASHES (And crossing line at completion of each lap)
6500	PANIC
7000	PIT STOP
7090	FUEL
7200	TYRE CHANGE
8100	FINISH (Successful completion of 5 laps)
8200	END ROUTINE
8250	HIT TREE
8300	HIT CROWD
8400	TYRE BURST
8500	TIME UP
8600	TANK EMPTY
8700	OFF SCREEN (if car gets too high up or too low down on the screen)
8999	TITLE

Listing 1

```
   1 REM 1234567890123456789
   2 REM 1234567890123456789012З
456
9500 LET J=0
9505 LET I=16514
9510 PRINT I;
9515 INPUT C
9520 PRINT TAB 15-LEN STR$ C;C;T
AB 18;CHR$ C;TAB 30-LEN STR$ (J+
C);J+C
9525 INPUT Z$
9530 IF Z$<>"Z" THEN GOTO 9515
9535 LET J=J+C
9540 POKE I,C
9545 IF I=16532 THEN LET I=16538
9550 IF I=16564 THEN GOTO 9600
9555 LET I=I+1
9560 GOTO 9510
9600 CLS
9605 LIST
9610 STOP
```

Listing 2

16514	42	E	42
16515	12	£	54
16516	64	RND	118
16517	14	:	132
16518	22	-	154
16519	6	▞	160
16520	32	4	192
16521	35	7	227
16522	126	?	353
16523	198	LEN	551
16524	128	█	679
16525	119	?	798
16526	16	(	814
16527	249	RAND	1063
16528	35	7	1098
16529	13	$	1111
16530	32	4	1143
16531	243	NEXT	1386
16532	201	TAN	1587
16539	42	E	1629
16540	12	£	1641

```
16541    64   RND     1705
16542    17   )       1722
16543   214   CHR$    1936
16544     2           1938
16545    25   ;       1963
16546   235   FOR     2198
16547    42   E       2240
16548    12   £       2252
16549    64   RND     2316
16550     1           2317
16551   181           2498
16552     2           2500
16553     9           2509
16554   237   GOSUB   2746
16555   184           2930
16556     1           2931
16557     0           2931
16558    32   4       2963
16559    35   7       2998
16560    54   Q       3052
16561     0           3052
16562    16   (       3068
16563   251   CLS     3319
16564   201   TAN     3520
```

Listing 3

```
  10 LET W=USR 16514
  20 FOR X=1 TO 20
  30 LET W=USR 16539
  40 NEXT X
  50 STOP
```

Listing 4

```
   1 REM E£RND -*477$4 NEXT TAN
   2 REM E£RND)CHR$ ; FOR E£RND
 GOSUB  470 ( CLS TAN
  50 REM
  55 GOSUB 9000
  60 LET A$=""
  90 LET FT=1000
 100 REM INSTRUCTIONS
 105 CLS
 110 PRINT AT 1,11;"ROAD RACE"
 120 PRINT AT 4,1;"YOU MUST COMP
LETE 5 LAPS IN AS  FAST A TIME A
S POSSIBLE USING   THE FOLLOWING
 KEYS :";AT 10,3;"5 LEFT";AT 10,
16;"5 AND DOWN";AT 11,16;"SHIFT
LEFT"
 130 PRINT AT 14,3;"8 RIGHT";AT
14,16;"8 AND DOWN";AT 15,16;"SHI
FT RIGHT";AT 18,3;"P PANIC";AT 1
8,16;"7 AND FAST";AT 19,16;"SHIF
T  UP"
 140 LET W=USR 16514
 150 PRINT AT 21,3;"PRESS ANY KE
Y TO CONTINUE";AT 21,3;"PRESS AN
Y KEY TO CONTINUE"
 160 IF INKEY$="" THEN GOTO 150
 200 CLS
 210 PRINT AT 10,1;"FUEL TANK CA
PACITY 200 LITRES";AT 12,3;"TYRE
 WEAR MAXIMUM 50"
 220 LET W=USR 16514
 230 PRINT AT 21,3;"PRESS ANY KE
Y TO CONTINUE";AT 21,3;"PRESS AN
Y KEY TO CONTINUE"
 240 IF INKEY$="" THEN GOTO 230
 250 CLS
 300 PRINT AT 5,1;"YOU USE UP MO
RE FUEL IF YOU HIT THE EDGE OR C
ARRY OUT MANOUVRES";AT 10,1;"YOU
R TYRES WEAR WHEN YOU PRESS  THE
 SHIFT KEY"
 310 LET W=USR 16514
 320 PRINT AT 21,3;"PRESS ANY KE
Y TO CONTINUE";AT 21,3;"PRESS AN
Y KEY TO CONTINUE"
 325 IF INKEY$="" THEN GOTO 320
 330 CLS
 350 PRINT AT 10,1;"YOU CAN RE-F
UEL OR CHANGE TYRES BY GOING TO
THE PITS, BUT TIME  GOES ON"
 360 LET W=USR 16514
 370 IF FT<1000 THEN PRINT AT 16
,3;"FASTEST TIME SO FAR ";FT;"SE
CONDS"
 380 PRINT AT 21,3;"PRESS ANY KE
Y TO CONTINUE";AT 21,3;"PRESS AN
Y KEY TO CONTINUE"
 390 IF INKEY$="" THEN GOTO 380
 400 CLS
1000 REM TRACK GRAPHICS
1010 DIM T$(44,32)
1020 LET T$(1)=""
1030 LET T$(2)=""
1040 LET T$(3)=""
1050 LET T$(4)=""
1060 LET T$(5)=""
1070 LET T$(6)=""
1080 LET T$(7)=""
1090 LET T$(8)=""
1100 LET T$(9)=""
1110 LET T$(10)=""
1120 LET T$(11)=""
1130 LET T$(12)=""
1140 LET T$(13)=""
1150 LET T$(14)=""
1160 LET T$(15)=""
1170 LET T$(16)=""
1180 LET T$(17)=""
1190 LET T$(18)=""
1200 LET T$(19)=""
1210 LET T$(20)=""
1220 LET T$(21)=""
1230 LET T$(22)="******"
1240 LET T$(23)="*************
***"
1250 LET T$(24)="************
*"
1260 LET T$(25)="***********"
1270 LET T$(26)="***********"
1280 LET T$(27)="***********
***"
1290 LET T$(28)="*********
****"
1300 LET T$(29)="*******
***"
1310 LET T$(30)="****
**"
```

```
1320 LET T$(31)="▒▒▒▒▒▒▒▒▒▒▒▒****
   **▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1330 LET T$(32)="▒▒▒▒▒▒▒▒▒▒▒▒****
   **▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1340 LET T$(33)="▒▒▒▒▒▒▒▒▒      **
   *▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1350 LET T$(34)="▒▒▒▒▒▒
     ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1360 LET T$(35)="▒▒▒▒▒▒
    ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1370 LET T$(36)="▒▒▒▒▒▒    ▒
   ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1380 LET T$(37)="▒▒▒▒▒▒    ▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1390 LET T$(38)="▒▒▒▒▒▒   ▒▒▒▒▒
▒▒▒▒****▒▒▒▒▒▒▒▒▒▒▒▒▒"
1400 LET T$(39)="▒▒▒▒▒   ▒▒▒▒▒
▒▒▒▒******▒▒▒▒▒▒▒▒▒▒▒"
1410 LET T$(40)="▒▒▒▒    ▒▒▒▒▒
▒▒▒▒******▒▒▒▒▒▒▒▒▒▒▒"
1420 LET T$(41)="▒▒▒    ▒▒▒▒▒▒
▒▒▒▒*****▒▒▒▒▒▒▒▒▒▒▒▒"
1430 LET T$(42)="▒▒    ▒▒▒▒▒▒▒
▒▒▒▒***▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1440 LET T$(43)="▒▒▒        ▒▒
▒▒▒▒**▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1450 LET T$(44)="▒▒▒▒▒
▒▒▒▒▒**▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
1480 CLS
1490 IF FT=1000 THEN GOTO 2900
1500 REM PRINT RECORD TIME
1510 PRINT AT 10,2;"RECORD TIME ";FT;" SECONDS"
1520 LET W=USR 16514
1530 PRINT AT 21,3;"PRESS ANY KEY TO CONTINUE";AT 21,3;"PRESS ANY KEY TO CONTINUE"
1540 IF INKEY$="" THEN GOTO 1530
2900 POKE 16418,0
2910 PRINT AT 22,0;" FUEL          LAPS    TYRE WEAR           TIME     "
2920 POKE 16418,2
3000 REM PRINT TRACK AND CAR AT THE START
3010 LET N=22
3020 LET E=0
3030 PRINT AT E,0;T$(N)
3040 LET N=N-1
3050 LET E=E+1
3060 IF E=21 THEN LET N=22
3070 IF E=21 THEN GOTO 3100
3080 GOTO 3030
3200 REM START VARIABLES
3210 LET LAPS=0
3220 LET FUEL=200
3230 LET G=10
3240 LET F=15
3260 PRINT AT G,F;"■"
3270 LET TW=0
3280 LET TIME=0
5000 REM CONTROL CENTRE
5050 LET W=USR 16539
5060 LET N=N+1
5070 PRINT AT 0,0;T$(N)
5080 IF N=44 THEN LET N=0
5090 IF G>19 THEN GOTO 8700
5100 IF CODE INKEY$=33 THEN GOTO 5500
5110 IF CODE INKEY$=36 THEN GOTO 5600
5120 IF CODE INKEY$=114 THEN GOTO 5700
5130 IF CODE INKEY$=115 THEN GOTO 5800
5140 IF CODE INKEY$=112 THEN GOTO 5860
5150 IF CODE INKEY$=53 THEN GOSUB 6500
5480 PRINT AT G,F;"■";AT G+1,F;" "
5485 GOSUB 6000
5490 GOTO 5900
5500 LET F=F-1
5510 PRINT AT G,F;"■";AT G+1,F+1;" "
5520 LET FUEL=FUEL-1
5560 GOSUB 6000
5570 GOTO 5900
5600 LET F=F+1
5610 PRINT AT G,F;"■";AT G+1,F-1;" "
5620 LET FUEL=FUEL-1
5660 GOSUB 6000
5670 GOTO 5900
5700 LET G=G+2
5710 LET F=F-2
5715 IF G>19 THEN GOTO 8700
5720 PRINT AT G-1,F+2;" ";AT G,F;"■"
5730 LET FUEL=FUEL-4
5740 LET TW=TW+2
5750 GOSUB 6000
5760 GOTO 5900
5800 LET G=G+2
5810 LET F=F+2
5815 IF G>19 THEN GOTO 8700
5820 PRINT AT G-1,F-2;" ";AT G,F;"■"
5830 LET FUEL=FUEL-4
5840 LET TW=TW+2
5850 GOSUB 6000
5855 GOTO 5900
5860 LET G=G-3
5865 IF G<1 THEN GOTO 8700
5870 PRINT AT G+4,F;" ";AT G,F;"■"
5880 LET TW=TW+1
5890 LET FUEL=FUEL-5
5900 REM PRINT FUEL ETC
5920 LET FUEL=FUEL-1
5930 LET TIME=TIME+1
5940 IF FUEL<0 THEN LET FUEL=0
5945 POKE 16418,0
5949 PRINT AT 22,6;"   "
5950 PRINT AT 22,6;FUEL;AT 22,29;LAPS;AT 23,11;TW;AT 23,29;TIME
5955 POKE 16418,2
5960 IF FUEL=0 THEN GOTO 8600
5970 IF LAPS=5 THEN GOTO 8100
5980 IF TIME=999 THEN GOTO 8500
5985 IF TW>49 THEN GOTO 8400
5990 GOTO 5000
6000 REM CHECK CRASHES
6010 LET N1=N
6020 IF N<=G THEN LET N1=N+44
6025 IF T$(N1-G,F+1)=" " THEN RETURN
6030 IF T$(N1-G,F+1)="▒" THEN LET FUEL=FUEL-20
6035 IF T$(N1-G,F+1)="▒" THEN LET TW=TW+5
6040 IF T$(N1-G,F+1)="*" THEN GOTO 8250
6050 IF T$(N1-G,F+1)="▧" THEN GOSUB 7000
6060 IF T$(N1-G,F+1)="▨" THEN GOTO 8300
6070 IF T$(N1-G,F+1)="▀" THEN LET LAPS=LAPS+1
6100 RETURN
6500 REM PAUSE
6520 PRINT AT 21,3;"PRESS ANY KEY TO CARRY ON";AT 21,3;"PRESS ANY KEY TO CARRY ON"
6530 IF INKEY$<>"" THEN GOTO 6520
6600 RETURN
```

```
7000 REM PIT STOP
7010 PRINT AT 20,0;" PRESS F FOR FUEL    T FOR TYRES ";AT 21,0;A$
7020 PRINT AT 20,7;"F";AT 20,20;"T"
7030 IF INKEY$="" THEN GOTO 7010
7040 IF CODE INKEY$=57 THEN GOTO 7200
7050 IF CODE INKEY$<>43 THEN GOTO 7010
7090 REM FUEL
7095 PRINT AT 20,0;A$;AT 21,0;A$
7100 PRINT AT 20,0;"HOW MANY LITRES ?  (APPROX    200)";AT 21,6;"YOU HAVE ";FUEL;" LEFT"
7110 INPUT Z$
7111 LET LZ=LEN Z$
7112 IF LEN Z$=0 THEN GOTO 7110
7113 FOR Z=1 TO LZ
7114 IF CODE Z$(Z)>37 OR CODE Z$(Z)<28 THEN GOTO 7100
7115 NEXT Z
7116 LET Z=VAL Z$
7120 POKE 16418,0
7130 PRINT AT 22,0;" FUEL                 LAPS    TYRE WEAR            TIME "
7140 PRINT AT 22,6;FUEL;AT 22,29;LAPS;AT 23,11;TW;AT 23,29;TIME
7145 LET Y=FUEL
7150 LET Y=Y+2
7155 LET TIME=TIME+1
7160 PRINT AT 22,6;Y;AT 23,29;TIME
7170 IF Y>=200 OR Y>=FUEL+Z THEN GOTO 7180
7175 GOTO 7150
7180 POKE 16418,2
7185 LET FUEL=Y
7190 IF FUEL>200 THEN LET FUEL=200
7193 PRINT AT 20,0;A$;AT 21,0;A$
7195 PRINT AT 20,1;"PRESS T FOR A TYRE CHANGE OR      PRESS ANY KEY";AT 20,7;"T"
7196 IF INKEY$="" THEN GOTO 7195
7197 IF CODE INKEY$<>57 THEN GOTO 8000
7200 REM TYRE CHANGE
7205 PRINT AT 20,0;A$;AT 21,0;A$
7210 PRINT AT 20,0;"THIS WILL TAKE ABOUT 20 SECONDS       PLEASE WAIT PATIENTLY"
7220 LET R=INT (RND*6)+18
7230 POKE 16418,0
7240 PRINT AT 22,0;" FUEL                 LAPS    TYRE WEAR            TIME "
7250 PRINT AT 22,6;FUEL;AT 22,29;LAPS;AT 23,11;TW;AT 23,29;TIME
7260 LET T=TIME
7270 LET T=T+1
7275 PRINT AT 23,29;T
7280 IF T=TIME+R THEN GOTO 7300
7290 GOTO 7270
7300 LET TIME=T
7310 LET TW=0
7320 PRINT AT 23,10;TW
7330 POKE 16418,2
8000 POKE 16418,0
8010 PRINT AT 22,0;" FUEL                 LAPS    TYRE WEAR            TIME "
8020 POKE 16418,2
8050 RETURN
8100 REM FINISH
8110 IF TIME<FT THEN LET FT=TIME
8120 CLS
8130 PRINT AT 10,1;"WELL DONE - YOU BEAT THE RECORD"
8140 LET W=USR 16514
8150 PRINT AT 15,4;"RECORD TIME ";FT;" SECONDS"
8160 PRINT AT 21,3;"PRESS ANY KEY TO CONTINUE";AT 21,3;"PRESS ANY KEY TO CONTINUE"
8170 IF INKEY$="" THEN GOTO 8160
8180 CLS
8200 REM END ROUTINE
8210 PRINT AT 10,4;"PRESS R FOR ANOTHER RACE";AT 12,4;"PRESS I FOR INSTRUCTIONS"
8215 LET W=USR 16514
8220 INPUT R$
8230 IF R$="R" THEN GOTO 1000
8235 IF R$="I" THEN GOTO 100
8240 GOTO 8220
8250 REM HIT TREE
8255 CLS
8260 PRINT AT 20,2;"YOU HAVE CRASHED INTO A TREE"
8270 GOTO 8200
8300 REM HIT CROWD
8305 CLS
8310 PRINT AT 20,2;"YOU HAVE KILLED A SPECTATOR"
8320 GOTO 8200
8400 REM TYRE BURST
8405 CLS
8410 PRINT AT 20,5;"YOU HAVE A BURST TYRE"
8420 GOTO 8200
8500 REM TIME UP
8505 CLS
8510 PRINT AT 20,4;"YOU HAVE RUN OUT OF TIME"
8520 GOTO 8200
8600 REM TANK EMPTY
8605 CLS
8610 PRINT AT 20,4;"YOU HAVE RUN OUT OF FUEL"
8620 GOTO 8210
8700 REM OFF SCREEN
8710 FOR E=8 TO 16
8720 PRINT AT E,0;"                                "
8725 NEXT E
8730 PRINT AT 10,3;"YOU HAVE RUN OUT OF SPACE"
8740 PRINT AT 14,3;"PRESS ANY KEY TO CONTINUE";AT 14,3;"PRESS ANY KEY TO CONTINUE"
8750 IF INKEY$="" THEN GOTO 8740
8760 GOTO 8180
8999 REM TITLE
9000 LET S$="[illegible]"
9010 LET V$="[illegible]"
9020 CLS
9030 LET W=USR 16514
9040 PRINT AT 4,0;S$;AT 11,0;V$
9050 PRINT AT 20,4;"PRESS ANY KEY TO CONTINUE";AT 20,4;"PRESS ANY KEY TO CONTINUE"
9060 IF INKEY$="" THEN GOTO 9050
9070 RETURN
9989 STOP
9990 SAVE "R[illegible]"
9995 RUN
```

Spectrum Disassembler

S. H. Man from Tottenham presents a program for taking apart the ROM or any machine code program all in 16K.

This BASIC program allows the budding machine code programmer to disassemble any section of memory. The clever part is that it fits into 16K!

All 694 of the Z80 instruction set are catered for and the program will PEEK at the addresses and convert the numbers to the correct mnemonic form. The program achieves this by storing all the mnemonics in DATA statements and only reading them as required, and not, as is usual, allocating them to an array.

Using the Program

When first RUN, the screen displays a request for the start address of the disassembly. This can be in either hex or decimal form and should be preceeded by either H or D as appropriate. For example, H0010 (address 16 in ROM) is a possible hex input, or D16 is the decimal version.

Note that leading zeros to make a four figure number MUST be added when entering in hex. Most invalid inputs have been trapped with an appropriate message.

The program will now disassemble from that address, displaying in hex and mnemonic form the contents of that and subsequent memory locations. The screen format shows the address in hex, the machine code and the mnemonic. When the screen is full you are given the message 'MORE ? (Y/N/C/E)'

Pressing Y clears the screen and disassembly continues.
Pressing N restarts the program.
Pressing C produces a COPY (provided that a printer is attached).
Pressing E ends the program.

You must press ENTER after selecting one of the above options. If you are using a 48K Spectrum then change 32767 to 65535 in line 980.



Figure 1. An example of the disassembly.

```
ØØØØ  F3       DI
ØØØ1  AF       XOR A
ØØØ2  11FFFF   LD DE,FFFF
ØØØ5  C3CB11   JP 11CB
ØØØ8  2A5D5C   LD HL,(5C5D)
ØØØB  225F5C   LD (5C5F),HL
ØØØE  1843     JR 43
ØØ1Ø  C3F215   JP 15F2
ØØ13  FF       RST ØØ38
ØØ14  FF       RST ØØ38
ØØ15  FF       RST ØØ38
ØØ16  FF       RST ØØ38
ØØ17  FF       RST ØØ38
ØØ18  2A5D5C   LD HL,(5C5D)
ØØ1B  7E       LD A,(HL)
ØØ1C  CD7DØØ   CALL ØØ7D
ØØ1F  DØ       RET NC
ØØ2Ø  CD74ØØ   CALL ØØ74
ØØ23  18F7     JR F7
ØØ25  FF       RST ØØ38
ØØ26  FF       RST ØØ38
ØØ27  FF       RST ØØ38
15F2  D9       EXX
15F3  E5       PUSH HL
15F4  2A515C   LD HL,(5C51)
15F7  5E       LD E,(HL)
15F8  23       INC HL
15F9  56       LD D,(HL)
15FA  EB       EX DE,HL
15FB  CD2C16   CALL 162C
15FE  E1       POP HL
15FF  D9       EXX
16ØØ  C9       RET
16Ø1  87       ADD A,A
16Ø2  C616     ADD A,16
16Ø4  6F       LD L,A
16Ø5  265C     LD H,5C
16Ø7  5E       LD E,(HL)
16Ø8  23       INC HL
16Ø9  56       LD D,(HL)
```

```
   9 REM ********************
  10 REM    **DISASSEMBLER***
  11 REM ********************
  12 DATA "B","C","D","E","H","L
","(HL)","A"
  15 DIM D$(8,4)
  17 FOR J=1 TO 8: READ D$(J): N
EXT J
  20 POKE 23658,8: REM CAPS LOCK
  40 REM **DEC TO HEX(D=Ø-16)**
  45 DEF FN A$(D)=CHR$ ((48+D AN
D D<1Ø)+(55+D AND D>9))
  47 DEF FN B(B$)=(CODE B$-48 AN
D CODE B$<64)+(CODE B$-55 AND CO
DE B$>64)
  5Ø INPUT "ADDRESS ?"; LINE k$:
 IF K$="" THEN GO TO 5Ø
  51 IF k$(1)<>"D" THEN GO TO 5
7
  52 IF LEN K$>6 OR K$="D" THEN
 GO TO 5Ø
  53 GO TO 15ØØ
  54 LET A=VAL (K$(2 TO )): GO T
O 6Ø
  55 REM
  57 IF K$(1)<>"H" OR LEN (K$)<>
5 THEN GO TO 5Ø
  58 GO TO 16ØØ
  59 GO SUB 13ØØ
  6Ø LET COUNT=Ø: CLS
  8Ø REM  ***START***
  9Ø LET C$="": LET B$=""
 1ØØ GO SUB 94Ø: GO SUB 98Ø
 11Ø IF C$(LEN C$-1 TO )="ED" TH
EN GO TO 51Ø
 12Ø IF C$(LEN C$-1 TO )="CB" TH
EN GO TO 55Ø
 13Ø IF C$(LEN C$-1 TO )="DD" TH
EN GO TO 59Ø
 14Ø IF C$(LEN C$-1 TO )="FD" TH
EN GO TO 61Ø
 15Ø REM  READ INSTRUCTION
 16Ø GO SUB 11ØØ
 185 LET N$=""
 19Ø GO SUB 1ØØØ: REM FIND N
 2ØØ IF LEN N$=Ø THEN GO TO 34Ø
 21Ø IF LEN N$=1 THEN GO TO 3ØØ
 23Ø GO SUB 98Ø
 24Ø GO SUB 98Ø
 25Ø LET B$=C$(LEN C$-1 TO )+C$(
LEN C$-3 TO LEN C$-2)
 27Ø LET N=FN B(N$(1))
 28Ø LET I$=I$( TO N-1)+B$+(I$(N
+1+(LEN N$=2) TO ) AND I$(LEN I$
)<>"#")
 29Ø GO TO 34Ø
 3ØØ REM LEN N$=1
 31Ø GO SUB 98Ø
 32Ø LET B$=C$(LEN C$-1 TO )
 33Ø GO TO 27Ø
 339 REM
 34Ø REM  ***PRINT ***
 35Ø PRINT A$;TAB (6);C$;TAB (15
);I$
 36Ø LET COUNT=COUNT+1
 37Ø IF COUNT=22 THEN GO TO 86Ø
 38Ø IF I$="RST ØØ28" THEN GO T
O 4ØØ
 39Ø GO TO 9Ø
 4ØØ REM  *** CAL ***
 41Ø LET C$=""
 42Ø GO SUB 94Ø: REM  A TO A$
 43Ø GO SUB 98Ø
 44Ø LET I$="DEFB "+C$
 46Ø PRINT A$;TAB (6);C$;TAB (15
);I$
 47Ø IF C$="38" THEN GO TO 36Ø
 48Ø LET COUNT=COUNT+1
 49Ø IF COUNT=22 THEN GO TO 86Ø
 5ØØ GO TO 41Ø
 51Ø REM  *** ED ***
 52Ø GO SUB 98Ø
 525 LET BYTE=BYTE-64
 53Ø RESTORE 25ØØ+1Ø*INT (BYTE/1
Ø)
 535 GO SUB 112Ø
 54Ø GO TO 17Ø
 549 REM
 55Ø REM   *** CB  ***
 553 GO SUB 98Ø
 555 GO SUB 56Ø
 556 GO TO 185
 56Ø RESTORE 3ØØØ+1Ø*INT (BYTE/6
4)
 565 LET V=INT (BYTE/8)-8*INT (B
YTE/64)
 57Ø FOR J=Ø TO V: READ L$: NEXT
 J
 571 LET V=BYTE-8*INT (BYTE/8)
 572 LET E$=D$(v+1)
 575 LET I$=L$+E$(1 TO 1+(3 AND
1+V=7))
 58Ø RETURN
 59Ø REM   ***  DD OR FD ***
 6ØØ LET X$="IX": GO TO 62Ø
 61Ø LET X$="IY"
 62Ø GO SUB 98Ø
 63Ø IF C$(LEN C$-1 TO )="CB" TH
EN GO TO 79Ø
 64Ø GO SUB 11ØØ
 67Ø REM  **FIND POSIT.OF HL**
 68Ø LET K=1
 69Ø IF K>LEN I$-1 THEN STOP
 7ØØ IF I$(K)="H" AND I$(K+1)="L
" THEN GO TO 72Ø
 71Ø LET K=K+1: GO TO 69Ø
```

```
 720 IF BYTE<=43 OR BYTE=57 OR B
YTE>=225 THEN  GO TO 760
 725 REM  ** HL > IX+D/IY+D **
 730 GO SUB 980
 735 LET U$=""
 736 IF K<>LEN I$-1 THEN  LET U$
=I$(K+2 TO )
 740 LET I$=I$(1 TO K-1)+X$+"+"+
C$(LEN C$-1 TO )+U$
 750 GO TO 185
 760 REM  ** HL -> IX OR IY **
 762 LET U$=""
 764 IF K<>LEN I$-1 THEN  LET U$
=I$(K+2 TO )
 770 LET I$=I$(1 TO K-1)+X$+U$
 780 GO TO 185
 790 REM  **FD CB OR DD CB**
 800 GO SUB 980: GO SUB 980
 810 LET BYTE=PEEK (A-1)
 820 GO SUB 560
 840 LET I$=I$( TO LEN I$-3)+X$+
"+"+C$(5 TO 6)+I$(LEN I$)
 850 GO TO 340
 860 REM  *** MORE? ***
 870 INPUT "MORE ?(Y/N/C/E)"; LI
NE Q$
 875 IF Q$="E" THEN  STOP
 880 IF Q$="N" THEN  GO TO 50
 882 IF Q$="Y" THEN  GO TO 890
 884 IF Q$<>"C" THEN  GO TO 860
 885 COPY : GO TO 860
 890 IF I$="RST 0028" THEN  GO T
O 910
 900 IF I$<>"DEFB "+C$ OR C$="38
" THEN  GO TO 60
 910 CLS : LET COUNT=0
 920 GO TO 400
 930 REM
 940 REM  **CONVERT A TO A$**
 950 LET A$="": LET R=A: LET K=3
 955 LET I=INT (R/(16^K))
 960 LET A$=A$+FN A$(I)
 965 LET R=R-(16^K)*I
 970 IF K=0 THEN  RETURN
 975 LET K=K-1: GO TO 955
 978 REM
 979 REM  *** SUBROUTINE **
 980 IF A>32767 THEN  GO TO 992
 981 LET BYTE=PEEK A
 983 LET I=INT (BYTE/16)
 985 LET C$=C$+FN A$(I)
 987 LET C$=C$+FN A$(BYTE-16*I)
 990 LET A=A+1: RETURN
 992 PRINT AT 21,6;"ADDRESS EXCE
ED MAXIMUN.": BEEP 3,50
 993 PRINT AT 21,6;"
           ": IF COUNT=0 THEN  G
O TO 50
 994 INPUT "COPY ?Y/N "; LINE Q$
 995 IF Q$<>"Y" THEN  GO TO 50
 996 COPY : GO TO 50
1000 REM  **FIND NO. OF #**
1005 LET N$=""
1010 FOR J=1 TO LEN I$
1020 IF I$(J)<>"#" THEN  GO TO 1
040
1030 LET N$=N$+FN A$(J)
1040 NEXT J
1050 RETURN
1099 REM  ** READ DATA **
1100 IF BYTE>=64 THEN  GO TO 114
0
1110 RESTORE 2010+10*INT (BYTE/1
0)
1120 FOR J=0 TO BYTE-10*INT (BYT
E/10): READ I$: NEXT J
1130 RETURN
1140 IF BYTE<=191 THEN  GO TO 11
80
1150 RESTORE 2200+10*INT ((BYTE-
192)/10)
1160 FOR J=0 TO BYTE-10*INT ((BY
TE-192)/10)-192: READ I$: NEXT J
1170 RETURN
1180 RESTORE 2700+10*INT ((BYTE-
64)/64)
1182 LET V=INT ((BYTE-64)/8)-8*I
NT ((BYTE-64)/64)
1184 FOR J=0 TO V: READ L$: NEXT
 J
1185 LET V=BYTE-64-8*INT ((BYTE-
64)/8)
1186 LET E$=D$(1+V)
1190 LET I$=L$+E$(1 TO 1+(3 AND
1+V=7))
1195 IF BYTE=118 THEN  LET I$="H
ALT"
1200 RETURN
1300 REM  ****HEX TO DEC ****
1305 LET A=0
1310 LET A$=K$(2 TO )
1320 FOR J=1 TO 4
1340 LET A=A+(16^(4-J))*FN B(A$(
J))
1345 NEXT J
1350 RETURN
1500 REM CHECK FOR SILLY ADDR.
1510 LET K=2
1530 IF CODE K$(K)<48 OR CODE K$
(K)>57 THEN  GO TO 50
1540 IF K=LEN K$ THEN  GO TO 54
1550 LET K=K+1: GO TO 1530
1600 REM CHECK FOR SILLY ADDR.
1610 LET K=2
```

```
1620 IF (CODE K$(K)>=48 AND CODE
 K$(K)<=57) OR (CODE K$(K)>=65 A
ND CODE K$(K)<=70) THEN  GO TO 1
635
1630 GO TO 50
1635 IF K=LEN K$ THEN  GO TO 59
1640 LET K=K+1: GO TO 1620
1999 REM
2000 REM  *** DATA ****
2010  DATA "NOP","LD BC,##","LD
(BC),A","INC BC","INC B","DEC B"
,"LD B,#","RLCA","EX AF,A'F'","A
DD HL,BC"
2020 DATA "LD A,(BC)","DEC BC","
INC C","DEC C","LD C,#","RRCA","
DJNZ #","LD DE,##","LD (DE),A","
INC DE"
2030 DATA "INC D","DEC D","LD D,
#","RLA","JR #","ADD HL,DE","LD
A,(DE)","DEC DE","INC E","DEC E"
2040 DATA "LD E,#","RRA","JR NZ,
#","LD HL,##","LD (##),HL","INC
HL","INC H","DEC H","LD H,#","DA
A"
2050 DATA "JR Z,#","ADD HL,HL","
LD HL,(##)","DEC HL","INC L","DE
C L","LD L,#","CP L","JR NC,#","
LD SP,##"
2060 DATA "LD (##),A","INC SP","
INC (HL)","DEC (HL)","LD (HL),#"
,"SCF","JR C,#","ADD HL,SP","LD
A,(##)","DEC SP"
2070 DATA "INC A","DEC A","LD A,
#","CCF"
2200 DATA "RET NZ","POP BC","JP
NZ,##","JP ##","CALL NZ,##","PUS
H BC","ADD A,#","RST 0000","RET
Z","RET "
2210 DATA "JP Z,##","","CALL Z,#
#","CALL ##","ADC A,#","RST 0008
","RET NC","POP DE","JP NC,##","
OUT (#),A"
2220 DATA "CALL NC,##","PUSH DE"
,"SUB #","RST 0010","RET C","EXX
","JP C,##","IN A,(#)","CALL C,#
#","***"
2230 DATA "SBC A,#","RST 0018","
RET PO","POP HL","JP PO,##","EX
(SP),HL","CALL PO,##","PUSH HL",
"AND #","RST 0020"
2240 DATA "RET PE","JP (HL)","JP
 PE,##","EX DE,HL","CALL PE,##",
"","XOR #","RST 0028","RET P","P
OP AF"
2250 DATA "JP P,##","DI","CALL P
,##","PUSH AF","OR H","RST 0030"
,"RET M","LD SP,HL","JP M,##","E
I"
2260 DATA "CALL M,##","***","CP
#","RST 0038"
2270 REM
2271 REM    *********
2300 REM
2499 REM   *** ED ***
2500 DATA "IN B,(C)","OUT (C),B"
,"SBC HL,BC","LD (##),BC","NEG",
"RET N","IM 0","LD I,A","IN C,(C
)","OUT (C),C"
2510 DATA "ADC HL,BC","LD BC,(##
)","","RET I","","LD R,A","IN D,
(C)","OUT (C),D","SBC HL,DE","LD
 (##),DE"
2520 DATA "","","IM 1","LD A,I",
"IN E,(C)","OUT (C),E","ADC HL,D
E","LD DE,(##)","",""
2530 DATA "IM 2","LD A,R","IN H,
(C)","OUT (C),H","SBC HL,HL","LD
 HL,(##)","","","","RRD"
2540 DATA "IN L,(C)","OUT (C),L"
,"ADC HL,HL","LD HL,(##)","","",
"","RLD","IN F,(C)",""
2550 DATA "SBC HL,SP","LD (##),S
P","","","","","IN A,(C)","OUT (
C),A","ADC HL,SP","LD SP,(##)"
2571 REM
2590 DATA "","","","","","","LDI
","CPI","INI","OUTI"
2600 DATA "","","","","LDD","CPD
","IND","OUTD","",""
2610 DATA "","","LDIR","CPIR","I
NIR","OTIR","","","",""
2620 DATA "LDDR","CPDR","INDR","
OTDR"
2699 REM   *** 64-191(dec)***
2700 DATA "LD B,","LD C,","LD D,
","LD E,","LD H,","LD L,","LD (H
L),","LD A,"
2710 DATA "ADD A,","ADC A,","SUB
 ","SBC A,","AND ","XOR ","OR ",
"CP "
2998 REM
2999 REM    **** CB ****
3000 DATA "RLC ","RRC ","RL ","R
R ","SLA ","SRA ","","SRL "
3010 DATA "BIT 0,","BIT 1,","BIT
 2,","BIT 3,","BIT 4,","BIT 5,",
"BIT 6,","BIT 7,"
3020 DATA "RES 0,","RES 1,","RES
 2,","RES 3,","RES 4,","RES 5,",
"RES 6,","RES 7,"
3030 DATA "SET 0,","SET 1,","SET
 2,","SET 3,","SET 4,","SET 5,",
"SET 6,","SET 7,"
```



FISH!

There's something fishy about Ken Rylett's Mancunian card game.

This simulation of the well known card game, which resembles "happy families", is played between you and the Spectrum, and the computer has obviously been tutored well.

Both you and the Spectrum are dealt a hand of cards and you then take turns to ask each other for cards of the same value, but you must hold at least one of those cards in your hand. For instance, if it is your turn and you have one or more Kings then you may ask the Spectrum for any Kings it has. When asked for any cards you must hand over all you have of that value. If the player asked does not have any cards of the value required, then the message "GOOD FISHING" is displayed and the cards are added to the asker's total. When the total of any card value is four then that player has a made a "book". The first player to reach seven books wins the game.

If a player asks for a card value which the opponent does not have then the message "GO FISH" is displayed and the player draws another card from the pack and play passes to the other player — unless the value of the card drawn happens to be the same as the card asked for originally, in this case the player continues his turn.

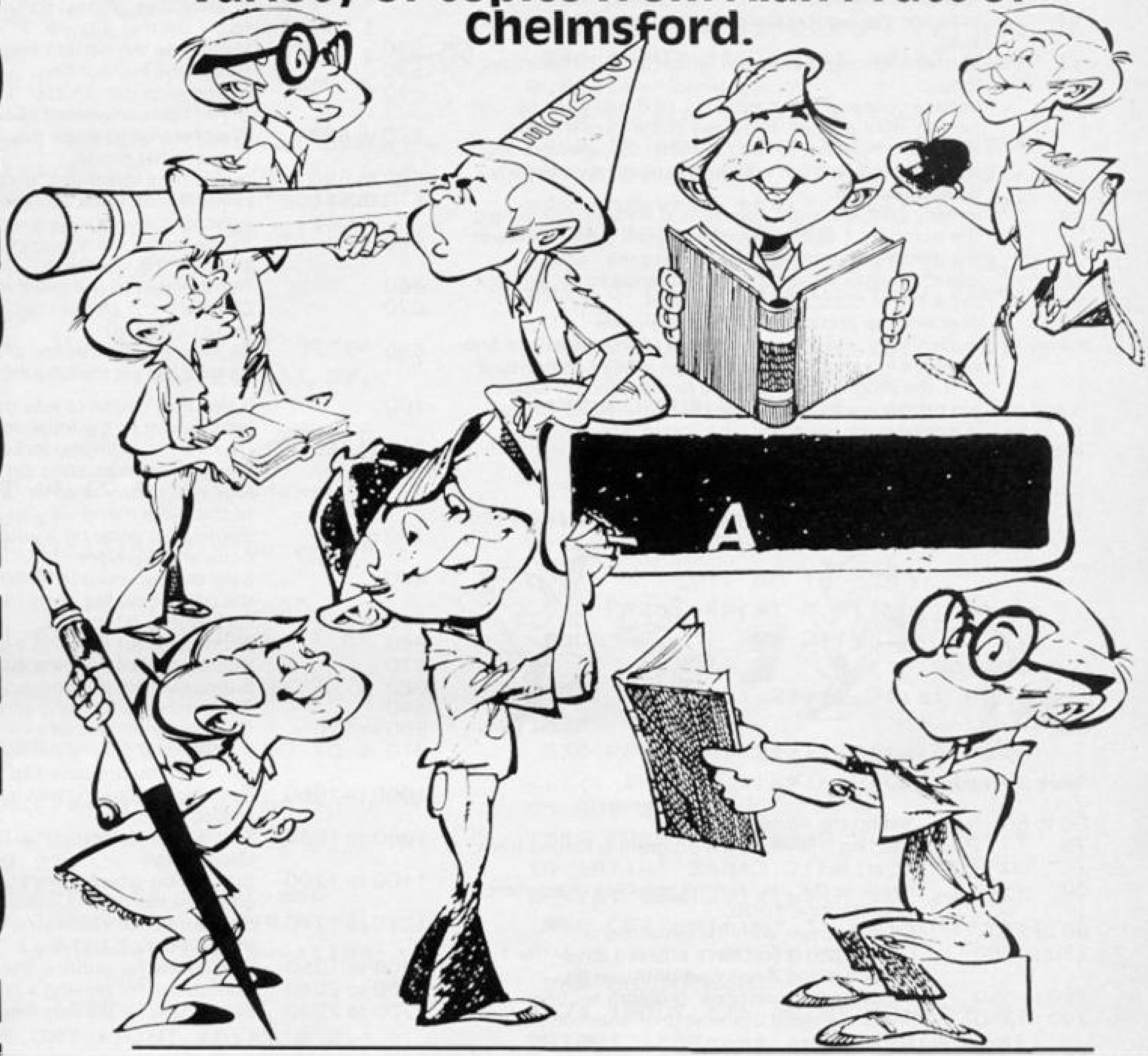
A fascinating game which has caused us much frustration in the office!

```
1000 LET Update=9500
1010 LET end=  9600
1020 GO SUB 9000
1030
1040 FOR n=1 TO 13
1050 LET D(n)     =n
1060 LET D(n+13)=n
1070 LET D(n+26)=n
1080 LET D(n+39)=n
1090 NEXT n
1100
1110 FOR n=1 TO 52
1120 LET a=INT (RND*52+1)
1130 LET b=INT (RND*52+1)
1140 IF  a=b THEN  GO TO 1130
1150 LET   c=D(a)
1160 LET D(a)=D(b)
1170 LET D(b)=c
1180 NEXT n
1190
1200 FOR n=1 TO 7
1210 LET a=D(n)
1220 LET P(a)=P(a)+1
1230 LET b=D(n+7)
1240 LET S(b)=S(b)+1
1250 NEXT n
1260
1270 BORDER 5: PAPER 5: INK 0:
    CLS
1280 PRINT  PAPER 6; BRIGHT 1;AT
 0,12;" FISH "
1290 PRINT  PAPER 2; BRIGHT 1; I
NK 7;AT 3,0;" PLAYER ";pb;AT 3,2
0;" SPECTRUM ";sb
1300 PRINT  PAPER 1; BRIGHT 1; I
NK 7;AT 8,3;" A 2 3 4 5 6 7 8 9
T J Q K "
1310 FOR n=9 TO 13
1320 PRINT  PAPER 6; BRIGHT 1;AT
 n,3;"
 "
1330 NEXT n
1340
2000 REM  PLAYER'S TURN
2010
2020 GO SUB Update
2030 INPUT  PAPER 0; INK 7; BRIG
HT 1;"WHICH CARD ? "; LINE A$
2040 IF A$(1)="T" THEN  LET c=10
: GO TO 2100
2050 IF A$(1)="J" THEN  LET c=11
: GO TO 2100
2060 IF A$(1)="Q" THEN  LET c=12
: GO TO 2100
2070 IF A$(1)="K" THEN  LET c=13
: GO TO 2100
2080 IF A$(1)="A" THEN  LET c=1
: GO TO 2100
2090 LET c=CODE A$-48: IF c<1 OR
 c>13 THEN  GO TO 2030
2100 IF P(c)=0 THEN  PRINT  BRIG
HT 1; PAPER 7; INK 2;AT 18,11; F
LASH 1;" CHEAT ": FOR n=1 TO 250
: NEXT n: PRINT AT 18,11;"
 ": GO TO 2030
2110 IF S(c)=0 AND top<53 THEN
PRINT AT 18,12; PAPER 2; INK 7;
BRIGHT 1; FLASH 1;" GO FISH ": F
OR n=1 TO 250: NEXT n: PRINT AT
18,12;"         ": GO TO 2500
2120 IF S(c)>0 THEN  PRINT AT 18
,9; PAPER 1; INK 7; BRIGHT 1; FL
ASH 1;" GOOD FISHING ": FOR n=1
TO 250: NEXT n: PRINT AT 18,9;"
              ": LET flag=1: LET
P(c)=P(c)+S(c): LET S(c)=0: LET
T(c)=1: GO SUB Update: IF P(c)=4
 THEN  LET pb=pb+1: LET T(c)=0:
GO SUB Update: GO SUB end: IF pb
=7 THEN  GO TO 9700
2130 IF flag=1 THEN  LET flag=0:
 GO TO 2030
2140 GO TO 3000
2500 LET d=D(top): LET top=top+1
: LET P(d)=P(d)+1: GO SUB Update
: IF P(d)=4 THEN  LET pb=pb+1: L
ET T(d)=0: GO SUB Update: GO SUB
 end:  IF pb=7 THEN  GO TO 9700
2510 IF c=d THEN  PRINT AT 18,9;
 PAPER 1; INK 7; BRIGHT 1; FLASH
 1;" GOOD FISHING ": FOR n=1 TO
250: NEXT n: PRINT AT 18,9;"
              ": GO TO 2030
2520 PRINT AT 18,8; PAPER 7; FLA
SH 1; BRIGHT 1;" You fished a ";
2530 IF d>1 AND d<11 THEN  PRINT
  PAPER 7; FLASH 1; BRIGHT 1;d;"
 ": GO TO 2700
2540 IF d=1    THEN  PRINT  PAPE
R 7; FLASH 1; BRIGHT 1;"ACE "  :
 GO TO 2700
2550 IF d=11   THEN  PRINT  PAPE
R 7; FLASH 1; BRIGHT 1;"JACK " :
 GO TO 2700
2560 IF d=12   THEN  PRINT  PAPE
R 7; FLASH 1; BRIGHT 1;"QUEEN ":
 GO TO 2700
2570 PRINT  PAPER 7; FLASH 1; BR
IGHT 1;"KING "
2700 FOR n=1 TO 250: NEXT n: PRI
NT AT 18,8;"
 "
3000 REM   SPECTRUM'S TURN
3010
```

```
3Ø2Ø LET e=1
3Ø3Ø IF T(e)=1 AND S(e)>Ø AND S(
e)<4 THEN  LET f=e: GO TO 3Ø6Ø
3Ø4Ø IF e<13 THEN  LET e=e+1: GO
 TO 3Ø3Ø
3Ø5Ø LET f=INT (RND*13)+1: IF S(
f)=Ø OR S(f)=4 THEN  GO TO 3Ø5Ø
3Ø6Ø PRINT  PAPER 2; INK 7; BRIG
HT 1; FLASH 1;AT 21,9;" HOW MANY
 ";
3Ø7Ø IF f=1  THEN  PRINT  PAPER
 2; INK 7; BRIGHT 1; FLASH 1;"AC
ES "  : GO TO 32ØØ
3Ø8Ø IF f=11  THEN  PRINT  PAPER
 2; INK 7; BRIGHT 1; FLASH 1;"JA
CKS " : GO TO 32ØØ
3Ø9Ø IF f=12 THEN  PRINT  PAPER
2; INK 7; BRIGHT 1; FLASH 1;"QUE
ENS ": GO TO 32ØØ
31ØØ IF f=13 THEN  PRINT  PAPER
2; INK 7; BRIGHT 1; FLASH 1;"KIN
GS " : GO TO 32ØØ
311Ø PRINT  PAPER 2; INK 7; BRIG
HT 1; FLASH 1;f;"'s "
32ØØ INPUT "               ";
 LINE A$
321Ø LET c=CODE A$-48: IF c<Ø OR
 c>13 THEN  GO TO 32ØØ
322Ø PRINT AT 21,9;"
     "
323Ø IF P(f)<>c THEN  PRINT AT 1
8,11; BRIGHT 1; FLASH 1;" CHEAT
": FOR n=1 TO 25Ø: NEXT n: PRINT
 AT 18,11;"        ": GO TO 3Ø6Ø
3235 IF c=Ø AND top<53 THEN  GO
TO 326Ø
324Ø LET S(f)=S(f)+P(f): LET P(f
)=Ø: LET T(f)=Ø: LET flag=1: GO
SUB Update: IF S(f)=4 THEN  LET
sb=sb+1: GO SUB Update: GO SUB e
nd: IF sb=7 THEN  GO TO 97ØØ
325Ø IF flag=1 THEN  LET flag=Ø:
 GO TO 3Ø2Ø
3255 GO TO 2ØØØ
326Ø LET c=D(top): LET top=top+1
327Ø LET S(c)=S(c)+1: GO SUB Upd
ate: IF S(c)=4 THEN  LET sb=sb+1
: GO SUB Update: GO SUB end: IF
sb=7 THEN  GO TO 97ØØ
3275 IF c<>f THEN  GO TO 2ØØØ
328Ø PRINT  PAPER 1; INK 7; BRIG
HT 1; FLASH 1;AT 18,9;" I FISHED
 A ";
329Ø IF c=1  THEN  PRINT  PAPER
1; INK 7; BRIGHT 1; FLASH 1;"ACE
 "     : GO TO 35ØØ
33ØØ IF c=11  THEN  PRINT  PAPER
 1; INK 7; BRIGHT 1; FLASH 1;"JA
CK "   : GO TO 35ØØ
331Ø IF c=12  THEN  PRINT  PAPER
 1; INK 7; BRIGHT 1; FLASH 1;"QU
EEN "  : GO TO 35ØØ
332Ø IF c=13  THEN  PRINT  PAPER
 1; INK 7; BRIGHT 1; FLASH 1;"KI
NG "   : GO TO 35ØØ
333Ø PRINT  PAPER 1; INK 7; BRIG
HT 1; FLASH 1;c;" "
35ØØ FOR n=1 TO 25Ø: NEXT n: PRI
NT AT 18,9;"                  ":
 GO TO 3Ø2Ø
9ØØØ REM   SET-UP
9Ø1Ø
9Ø2Ø DIM P(13): REM Player's
9Ø3Ø DIM S(13): REM Spectrum's
9Ø4Ø DIM D(52): REM Deck
9Ø5Ø DIM T(13): REM Spec's loses
9Ø6Ø DIM A$(2): REM Request
9Ø7Ø
9Ø8Ø LET top=15: LET flag=Ø
9Ø9Ø LET  pb=Ø : LET   sb=Ø
91ØØ POKE 23658,8
911Ø RETURN
92ØØ
95ØØ REM    UPDATE
951Ø
952Ø PRINT  PAPER Ø; INK 7; BRIG
HT 1;AT 11,3;" ";P(1);" ";P(2);"
 ";P(3);" ";P(4);" ";P(5);" ";P(
6);" ";P(7);" ";P(8);" ";P(9);"
";P(1Ø);" ";P(11);" ";P(12);" ";
P(13);" "
953Ø PRINT  PAPER 2; INK 7; BRIG
HT 1;AT 3,8;pb;AT 3,3Ø;sb
954Ø RETURN
955Ø
96ØØ REM   CHECK FOR FINISH
961Ø
962Ø LET check1=Ø: LET check2=Ø
963Ø FOR l=1 TO 13
964Ø IF S(l)>Ø AND S(l)<4 THEN
 LET check1=1
965Ø IF P(l)>Ø AND P(l)<4 THEN
 LET check2=1
966Ø NEXT l
967Ø IF check1>Ø AND check2>Ø
 THEN  RETURN
968Ø
97ØØ REM   END OF GAME
971Ø
972Ø INPUT  PAPER Ø; INK 7; BRIG
HT 1;"ANOTHER GAME ?"; LINE A$
973Ø IF A$(1)="Y" THEN  GO SUB 9
ØØØ: GO TO 1Ø4Ø
974Ø STOP
```

Ask Me Another

A great way of testing knowledge on a variety of topics from Alan Pratt of Chelmsford.



This program is useful as an educational aid for anyone with young children. The object is to find the answers to eight questions from a jumbled list of those answers.

When typing in the program the graphics characters used for the two faces are as follows: Top half of both faces 'AB', bottom half of the smiling face 'CD', bottom half of the sad face 'EF'.

When you RUN the program the screen is filled with a menu and associated graphics display. You will need to press the '2' key initially to provide a set of questions and answers. It is not necessary to press the ENTER key in response to menu selections in this program except when entering questions and answers or editing a set. The screen now fills with instructions and a list of which of the five sets are full and which empty. At this stage all five will be empty but later you may wish to replace a set or edit specific questions and answers. This is possible by selecting a set which is full. Editing is useful for correcting mistakes or changing single questions.

Having selected which set you wish to fill, then I suggest you continue in the absence of the children, since the questions and answers are printed in the correct order. Enter eight questions and answers in response to the instructions. Once complete, check for spelling mistakes. If there are any you will need to return to this section, select the same set and use 'e' for edit. Press any key to return to the initial menu. Now try pressing key '3'. The Spectrum will ask you to set the tape to record. Record the program and then LOAD it back in (after first verifying it). Once loaded it will run automatically and all the sets of questions and answers you entered will be available.

Now to the main purpose of the program. Press key '1' when the menu is displayed and a list of the sets will be printed. Select a set which is full and the questions will be printed in order with the answers jumbled. Each qestion may be selected only once in any order and it is up to your children to try and find the correct answer. Remember, each time you fill a new set or change an existing set the END routine will ask you to record the program.

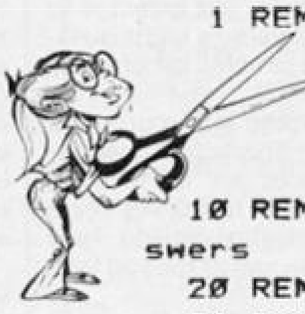
Figure 1. Variables List.

l$	contains the titles for each of the five sets of questions and answers.
q$	contains the eight questions which are 28 characters long.
a$	contains the eight answers which are 13 characters long.
p$	initially contains the numbers 1 to 8 and ends up containing eight spaces. It is used in the 'Answer the questions' routine, firstly to jumble the answers and secondly to establish which questions have already been answered.
c$	initially contains eight spaces and ends up containing the numbers 1 to 8 in a jumbled order in the 'Answer the questions' routine.
t$	initially contains five '0' which change to '1' when a set is full of questions and answers.
e$	contains the present keyboard response.
new	= 0 initially and is set to 1 if any of the questions and answers are changed or a new set is filled. When new ' = 1 the program needs to be re-recorded on tape.
x and y	are temporary variables used in FOR–NEXT loops.
i	is a temporary variable.
n,q,a	= number between 1 and 8 in response to the appropriate key pressed.
r	= random integer between 1 and 8.
t	= counter incremented from 0 to 8. When t = 8 you have answered all the questions.

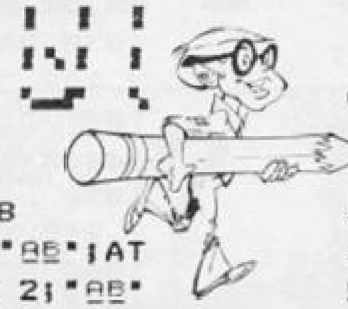


Figure 2. Program Lines.

50 to 60	Set up the variables.
70	Starting line when the program is loaded from tape.
80	Reads in the data for the graphics characters in subroutine 1000.
90 to 220	Fill the screen with the initial menu.
230 to 260	Look to see if you have pressed either the 1, 2 or 3 keys and if not wait until you do.
280 to 590	'Answer the questions' program section.
290 to 310	Prints the state of the sets of questions and answers.
320	Waits for you to make a valid choice and puts this into the variable 'n'.
330	Checks if your choice of set is empty. If so then a comment is printed and you are returned to the initial menu.
340 to 390	This routine jumbles the numbers 1 to 8 from p$ and puts them into c$. In this way the computer keeps a record of which answers belong to which questions.
400	Draws lines on the screen and prints the title for the set.
410 to 420	Print the questions in order and the answers jumbled according to the numbers in c$.
430	Initialises p$ and the variable 't'.
440 to 450	Ask you to select a question by number and put the response in 'q'.
460	Checks if your choice has already been chosen. If so a comment is printed and you return to LINE 440.
470	Highlights your question.
480 to 490	Ask you to select an answer by number and puts the response in 'a'.
500	Shows which answer you chose by flashing the number.
510	Prints a comment and the smiling face if the answer was correct.
520	Prints a comment and the sad face if the answer was wrong. Returns you to LINE 480 to try another answer.
530	Highlights the correct answer.
540	Clears the highlighting.
550	Increments the counter 't'. If this has reached 8 you have answered all the questions.
570 to 590	Wait for you to press any key and returns you to the initial menu.
600 to 890	'Enter new questions' program section.
610 to 640	Prints instructions and the state of the sets of questions and answers.
650	Waits for you to make a valid choice and puts this into the variable 'n'.
660	Returns you to the initial menu if n = 6.
670	Checks if the chosen set is empty. If it is then go to LINE 790.
690	By selecting 'r' this line allows you to replace an existing set completely.
700	If you don't wish to edit the set this line returns you to the initial menu.
710 to 780	Routine for changing individual questions and answers in an existing set. This routine is repeated until you enter '9' which returns you to the initial menu.
790 to 820	Routine for entering a whole set of eight questions and answers.
820	Sets the appropriate '0' in t$ to a '1' to show the new set is full. Sets new = 1.
830 to 850	As LINES 570 to 590.
860	Subroutine for entering a question.
870	Subroutine for entering an answer.
880	Subroutine for printing a question.
890	Subroutine for printing an answer.
900 to 920	'End' program section.
910	If any of the sets have been changed or a new set filled you are asked to record the program.
1000 to 1050	Data for the smiling and sad faces graphics characters.
1060 to 1090	Subroutine for setting up the graphics characters
1100 to 1200	Subroutine which draws lines on the screen to separate the title, questions and answers.
1210 to 1240	Subroutine for identifying a key press between the numbers 1 and 8.
1300 to 1350	Subroutine for printing the state of the sets.
2000 to 2040	Subroutine for playing a happy tune.
2200 to 2230	Subroutine for playing a sad tune.



```
   1 REM ************************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         ************************
  10 REM quiz - Questions and An
swers
  20 REM
  50 DIM l$(5,32): DIM q$(5,8,28
): DIM a$(5,8,13): DIM p$(8): DI
M t$(5): DIM e$(1)
  60 LET t$="00000"
  70 BORDER 7: PAPER 7: INK 0: C
LS
  80 RESTORE 1000: GO SUB 1000:
LET new=0
```



```
  90 REM Initialising
 100 DIM c$(8): CLS
 110 PRINT AT 1,8; INK 1;"▗▀▀▖ ▐
 ▐ ▐ ▀▀▜           ▐ ▐ ▐
 ▐ ▐  ▗▘           ▐▚▐ ▐
 ▐ ▐ ▗▘            ▝▄▟ ▚
▄▟ ▐ ▙▄▄ ▝
 120 FOR y=1 TO 13 STEP 6
 130 FOR x=1 TO 29 STEP 28
 140 PRINT AT y,x; INK 4;"AB";AT
 y+1,x;"CD";AT y+3,x; INK 2;"AB"
;AT y+4,x;"EF"
 150 NEXT x
 160 NEXT y
 170 PRINT  INK 4;AT 19,1;"AB";A
T 20,1;"CD";AT 19,29;"AB";AT 20,
29;"CD"
 180 PRINT AT 8,7;"Select your o
ption :"
 190 PRINT AT 11,4;"1 = Answer t
he questions"
 200 PRINT AT 13,4;"2 = Enter ne
w questions"
 210 PRINT AT 15,4;"3 = End"
 220 PRINT AT 19,5; BRIGHT 1;"Pr
ess the correct key"
 230 IF INKEY$="1" THEN  GO TO 2
80
 240 IF INKEY$="2" THEN  GO TO 6
00
 250 IF INKEY$="3" THEN  GO TO 9
00
 260 GO TO 220
 280 REM Answer the questions
 290 CLS : INPUT "": PRINT AT 0,
7; BRIGHT 1;"Answer the question
s"
 300 GO SUB 1300
 310 PAUSE 30: PRINT #0;AT 0,0;"
Enter the set you require ?"
 320 GO SUB 1210: LET n=i: IF n>
5 THEN  GO TO 320
 330 IF t$(n)="0" THEN  PRINT AT
 6,8; FLASH 1;"That set is empty
": INPUT "": PAUSE 150: GO TO 90
 340 CLS : LET p$="12345678"
 350 FOR x=1 TO 8
 360 LET r=INT (RND*8)+1: IF p$(
r)<>" " THEN  GO TO 380
 370 GO TO 360
 380 IF x=r THEN  GO TO 360
 390 LET c$(x)=p$(r): LET p$(r)=
" ": NEXT x
 400 GO SUB 1100: PRINT  INK 1;A
T 0,0;l$(n): FOR x=1 TO 8
 410 PRINT AT 2*x+2,0;x;AT 2*x+2
,2;q$(n,x)(1 TO 14);AT 2*x+3,2;q
$(n,x)(15 TO 28);AT 2*x+2,17;a$(
n,VAL c$(x));AT 2*x+2,31;x
 420 NEXT x
 430 LET p$="12345678": LET t=0
 440 INPUT "": PRINT #0;AT 0,0;"
Question number ?"
 450 GO SUB 1210: LET q=i
 460 IF p$(q)=" " THEN  PRINT #0
;AT 0,0; BRIGHT 1;"You have alre
ady answered that": BEEP 1,-12:
PAUSE 50: GO TO 440
 470 PRINT  BRIGHT 1;AT 2*q+2,2;
q$(n,q)(1 TO 14);AT 2*q+3,2;q$(n
,q)(15 TO 28)
 480 INPUT "": PAUSE 30: PRINT #
0;AT 0,0;"Answer number ?"
 490 GO SUB 1210: LET a=i
 500 PRINT AT 2*a+2,31; FLASH 1;
a: PAUSE 50
 510 IF VAL c$(a)=q THEN  PRINT
#0;AT 0,0;"You got it right    "
; INK 4;"AB
         CD": GO TO 530
 520 PRINT #0;AT 0,0;"No that is
 wrong     "; INK 2;"AB
                         EF": GO SUB
2200: PRINT AT 2*a+2,31;a: GO TO
 480
 530 PRINT AT 2*a+2,31;a;AT 2*a+
2,17; BRIGHT 1;a$(n,VAL c$(a)):
GO SUB 2000
 540 PRINT AT 2*q+2,2;q$(n,q)(1
TO 14);AT 2*q+3,2;q$(n,q)(15 TO
28);AT 2*a+2,17;a$(n,VAL c$(a))
 550 LET p$(q)=" ": LET t=t+1: I
F t=8 THEN  GO TO 570
 560 GO TO 440
 570 INPUT "": PRINT #0;AT 0,0;
BRIGHT 1;"Press any key to conti
nue"
 580 PAUSE 0
 590 GO TO 90
 600 REM Enter new questions
 610 CLS : PRINT AT 0,7; BRIGHT
1;"Enter new questions"
 620 PRINT AT 2,0;"This program
will hold up to 5  sets of quest
ions and answers.  Each set is i
dentified by a    "
 625 PRINT AT 5,0;"number 1 to 5
. This section      tells you whi
ch sets are full    and allows yo
u to fill or        change any se
t with new          questions and
 answers."
 630 GO SUB 1300: PRINT AT 19,3;
"Press 6 to leave this routine"
```



```
640 INPUT "": PRINT #0;AT 0,0;"Which Set do you wish to fill ?"
650 GO SUB 1210: LET n=i: IF n>6 THEN GO TO 650
660 INPUT "": IF n=6 THEN  GO TO 90
670 IF t$(n)="0" THEN  GO TO 790
680 INPUT "Edit or Replace - ENTER e or r  (or any other key to Exit)  "; LINE e$
690 IF e$="r" THEN  GO TO 790
700 IF e$<>"e" THEN  GO TO 90
710 CLS : GO SUB 1100: PRINT  INK 1;AT 0,0;l$(n)
720 FOR x=1 TO 8: GO SUB 880: GO SUB 890: NEXT x
730 INPUT "Which question do you wish to   change ? (Enter 9 to leave) ";x
740 LET x=INT x: IF x<1 OR x>9 THEN  GO TO 730
750 IF x=9 THEN  GO TO 90
760 PRINT AT 2*x+2,0; BRIGHT 1;x: GO SUB 860
770 PRINT AT 2*x+2,31; BRIGHT 1;x: GO SUB 870
780 LET new=1: GO TO 730
790 CLS : INPUT "Title (max 32 letters) ?", LINE l$(n)
800 GO SUB 1100: PRINT  INK 1;AT 0,0;l$(n)
810 FOR x=1 TO 8: GO SUB 860: GO SUB 870: NEXT x
820 LET t$(n)="1": LET new=1
830 INPUT "": PRINT #0;AT 0,0; BRIGHT 1;"Press any key to continue"
840 PAUSE 0
850 GO TO 90
860 INPUT "Question (max 28 letters) ?", LINE q$(n,x): GO SUB 880: RETURN
870 INPUT "Answer (max 13 letters) ? ", LINE a$(n,x): GO SUB 890: RETURN
880 PRINT AT 2*x+2,0;x;AT 2*x+2,2;q$(n,x)(1 TO 14);AT 2*x+3,2;q$(n,x)(15 TO 28): RETURN
890 PRINT AT 2*x+2,17;a$(n,x);AT 2*x+2,31;x: RETURN
900 REM End of program
910 IF new THEN  SAVE "quiz" LINE 70: PAUSE 30
920 CLS : STOP
1000 DATA "a",7,31,48,96,76,204,192,193
1010 DATA "b",224,248,12,6,50,51,3,131
1020 DATA "c",193,192,216,79,99,48,31,7
1030 DATA "d",131,3,27,242,198,12,248,224
1040 DATA "e",193,192,195,71,108,48,31,7
1050 DATA "f",131,3,195,226,54,12,248,224
1060 FOR x=1 TO 6: READ e$
1070 FOR y=0 TO 7
1080 READ i: POKE USR e$+y,i
1090 NEXT y: NEXT x: RETURN
1100 REM Lines
1102 PLOT 0,164: DRAW 255,0
1104 PLOT 0,163: DRAW 255,0
1110 PLOT 0,148: DRAW 255,0
1120 PLOT 0,147: DRAW 255,0
1130 PLOT 0,10: DRAW 255,0
1140 PLOT 0,11: DRAW 255,0
1145 PLOT 0,148: DRAW 0,15
1150 PLOT 131,11: DRAW 0,152
1160 PLOT 132,11: DRAW 0,152
1165 PLOT 255,148: DRAW 0,15
1170 PLOT 11,11: DRAW 0,136
1180 PLOT 244,11: DRAW 0,136
1190 PRINT  INK 1;AT 2,4;"QUESTIONS";AT 2,20;"ANSWERS"
1200 RETURN
1210 LET e$=INKEY$
1220 LET i=CODE e$-48
1230 IF i>0 AND i<9 THEN  RETURN

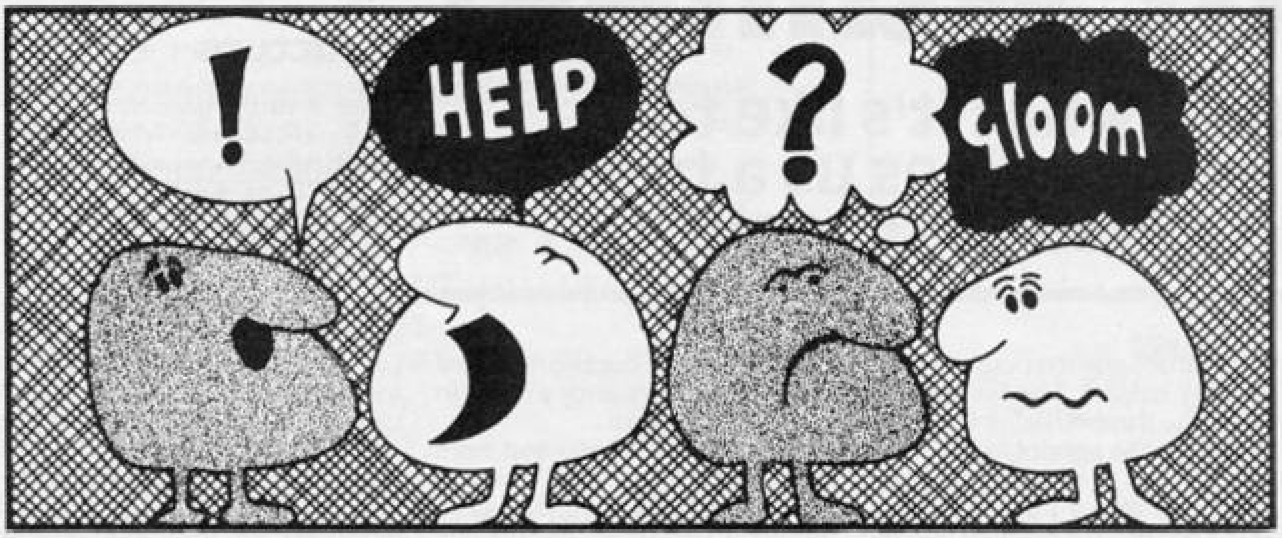
1240 GO TO 1210
1300 REM State of the data
1310 PRINT AT 11,8; BRIGHT 1;" SET No.    STATE "
1320 FOR i=1 TO 5
1330 PRINT AT i+12,11;i;AT i+12,19;("FULL" AND t$(i)="1")+("EMPTY" AND t$(i)="0")
1340 NEXT i
1350 RETURN
2000 REM Got it right
2010 LET w=0.07
2020 BEEP 3*w,12: BEEP w,16: BEEP 2*w,14: BEEP w,17: BEEP 3*w,16: BEEP 5*w,12
2030 PAUSE 100
2040 RETURN
2200 REM Got it wrong
2210 LET w=0.1
2220 BEEP 1.5*w,7: BEEP w,4: BEEP 1.5*w,0: BEEP w,4: BEEP .8*w,2: BEEP 2*w,2
2230 RETURN
```







Problem Page



Dear David,
I have just acquired a monitor which I want to connect to my Sinclair Spectrum. I tried putting the signal from the regular TV outlet of my computer to the monitor, but it did not work. What do I so now? Do I have to connect the rear part of the computer using a lead of some sort, or do I have to use an interface?

Amit Garg,
Panipat, India

Dear Amit,
Both your suggestions are correct, but I'd advise you to use the second idea, of an interface. First of all though check that your monitor is fitted with a regular RGB socket, otherwise you'll need to get further advise, depending in the type of signal input on your monitor. You can obtain an RGB signal via the expansion port of a Spectrum, but it will mean some internal soldering; so that's not advisable unless you're an expert, and you don't mind losing the guarantee on you computer. An interface is a safer option. RGB interfaces vary in price; one offering good value for money comes from Adapt Electronics (20 Starling Close, Buckhurst Hill, Essex) and costs £35.

Bob Brown of Maidenhead sent in a similar question; I hope my reply helps both of you.

TS-1000

Dear David,
I have a TS-1000, the Brazilian version of the ZX81. All the ZX81 software I have works well on the machine, so, apart from a build-in joystick port, my micro appears identical to a ZX81. However, the keys the joysticks operates are fixed and cannot be altered to suit different games. They are fixed to the cursor keys (5-8), and zero for fire. I'd like to translate my programs, which use other keys for movement, to use joysticks. I have no trouble with BASIC programs, but tracking down cursor control commands in a machine code listing is something I know little about. Can you help?

Mark Welch,
Rio Bonito, Brazil

Dear Mark,
It will be a major undertaking, but if you're prepared to be patient, then it can be done. First, you'll need to obtain an assembly language listing; that could be many pages long — a printer will help. If you've been following my machine code series, then you may remember that one way to check if a key is pressed is to use a ROM routine; it would be CALLed from a machine code routine, and it resides at 02BB (hex.) The value obtained might then be translated to a ZX81 character through another CALL to ROM, at address 07BD. The keypress can then be checked through a number of CP (compare) instructions, and it is the comparison values that you will have to alter.

TS-2068

Dear David,
I own a Timex 2068, the American Spectrum. There are two questions I have on string handling. First of all, the Spectrum handles both upper and lower case strings as the same string. Is there any way to make an upper case string different from a lower string, to allow 52 variables rather than 26?

My second question: When I DIM a string array, all the strings are the same length. So, when I print out strings from an array, can 'print' unwanted spaces, where the number of letters is less than the dimension of the array. Can the strings be cut down, so that I can overcome this problem?

Eric Ferguson,
Suisun City, CA, USA.

Dear Eric,
The answer to your first question is no, it can't be done; if you find the limit of 26 string variables a problem then use arrays more often. . . which brings me into your second question.

You may find the following subroutine useful for shortening strings. This routine assumes your array is a$, and you want element 'x' from that array:

```
100 LET z$ = a$(x)
110 FOR i = LEN z$ to 2 STEP -1
120 IF z$(i)<>" " THEN GOTO 150
130 LET z$ = z$( TO LEN z$ - 1)
140 NEXT i
150 PRINT z$;
160 RETURN
```

The FOR-NEXT loop gradually shortens z$ (into which the string array element was placed) from the right-hand end, until the first character is encountered. z$ will, therefore, contain no 'wasted' spaces.

Paper, Paper, Ink

Dear David,
I'm trying to write a program on my Spectrum which uses the INK and PAPER commands; the effects that I'm getting are not what I expected. For example, I want to change the PAPER colour on the screen, leaving the display intact, but using PAPER has no effect. I'm sure that it's because I don't understand how to use these commands. Please could you explain them.

Peter Thompson,
Stoke-on-Trent.

Dear Peter,
There are two ways to use INK and PAPER. The easiest way is in setting the GLOBAL colours. That is, once you set a colour, e.g. INK 5: PAPER 7, you get those colours everytime you use PRINT or the drawing commands. You won't see the change in colour straightaway after using INK and PAPER (unless you CLS, when the whole screen, except the bottom two lines, will be in your selected PAPER colour). That's a common misunderstanding, and, that's why you are not getting the effects you hope for.

Using the INK and PAPER commands in a PRINT statement will cause a temporary change in colour. The new colours will only last while that particular PRINT is being carried out, after which, the global colours are returned.

To change the screen colours without removing the characters on the screen involves POKEing suitable values into the attribute file. Try experimenting with the following routine to get an idea of how it's done:

```
10 INPUT "Enter a number, 0-255";a
20 FOR i=22528 TO 23295
30 POKE 1,a
40 NEXT i
50 GOTO 10
```

Generally, you will only need to input numbers between 0 and 63, although larger numbers also produce interesting effects. ■

Attack of the Mutant Wallies!

Ever wondered what it's like to be a teacher? A. Haslam gives us a taste of life in a Walsall school!

Teachers are fighting for a pay rise, but have you ever wondered what they are doing to deserve one? Well here's your chance to try out a typical day in a typical school.

Problems? You can't expect to succeed at the first attempt, so you had better report your problems to the Head, he's sure to help! But of course you have to get there and the pupils are not exactly helpful. The first screen sees you trapped behind a pile of desks by the school Idiot, using keys 5 and 8 to move left and right, get into a position where you can throw a book at him by pressing key 0. Hit him and you will escape into . . . .

The corridor, where lurks the school Burk. He immediately locks you in the cupboard where you must hit him with a pellet in order to escape to . . . .

The entrance hall, and here life becomes difficult. For here you meet the Super Wally who is throwing sweets across the hall in a straight line, you have to find a gap and get across to reach the Super Wally alarm bell. Reach it and the Head will come out to remove Super Wally and punish him as he deserves.

Oh, one last thing, as it is very near the end of the day you only have a limited time to do each of these things before school ends.

Structure

Lines 3 to 60 give the instructions, these are "comprehensive"

Lines 70 to 490 are the first level, the classroom.

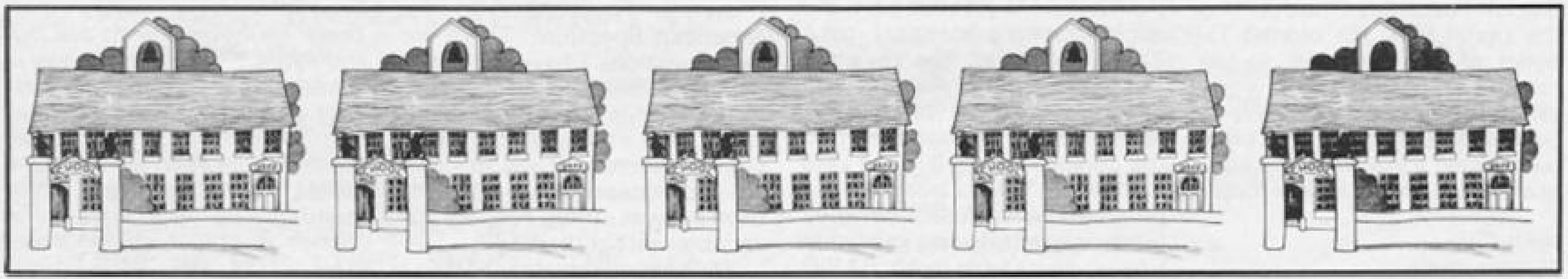
Lines 495 to 820 are the corridor.

Lines 830 to 1070 are the Entrance hall routines.

Lines 1080 to 8130 Winning routines and High score.

Lines 9000 to 9400 graphics and tune.

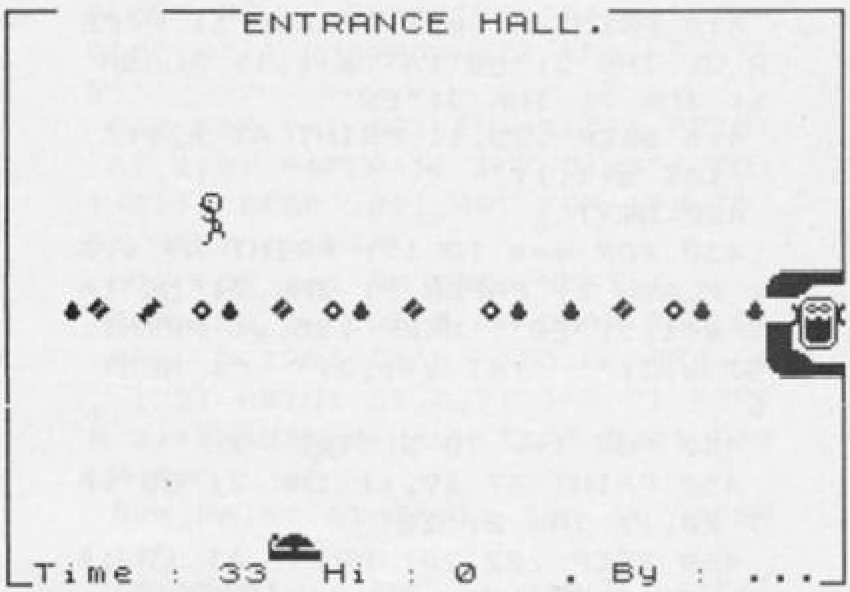
A special point of interest is the use of the UDGs. These are redefined no less than FOUR times throughout the run of the program. So, Mr. Chips, take cane in hand and NUT insurance in the other and do your duty!





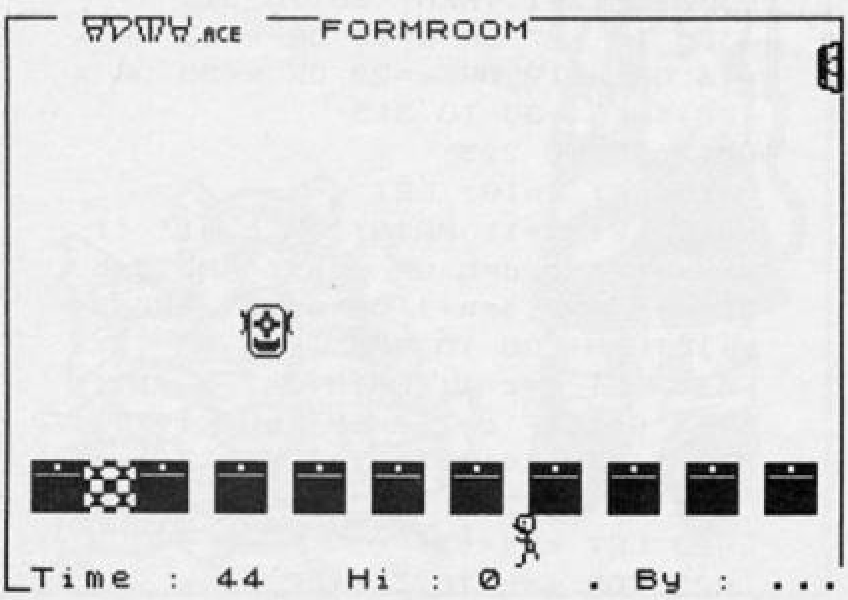

```
   1 REM ************************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         ************************
   3 RESTORE 9000: FOR i=144 TO
144: FOR n=0 TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
   5 BRIGHT 1: PAPER 0: INK 7: B
ORDER 0: CLS
   6 LET hi=0: LET a$="..."
  10 LET o$="
                 A T T A C K   O
F   T H E   M U T A N T   W A L
L I E S   .   .   .   .   .
.   .   Guide yourself past the
'IDIOTS'  ,  'BURKS'  ,  and DRE
ADED , HORRIBLE , SUPER WALLY  !
  !  !  !  .   .   ."
  20 LET o$=o$+"   .   .   You m
ust get out of the Formroom , Go
 down the corridor , And get thr
ough the Entrance Hall to get to
 the Headmaster's study to tell
him how bad they have been."
  30 LET o$=o$+"   .   .   .   Y
```

```
ou use '5' to move Left   , '8'
to move right   ,  '7' to move up
 ,  '6' to move down and '0' to
 fire.    .    .    .    .   Goo
d Luck  (You'll need it !!!)
   Press any key to start   "
  35 PRINT AT 0,0; INK 2; PAPER
7;"AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA";AT 2,0; INK 2; PAPER 7;"AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
  40 FOR i=1 TO LEN o$-31: PRINT
 AT 1,0; INK 1; PAPER 6;;o$(i TO
 i+31): FOR n=0 TO 10: NEXT n: B
EEP .001,40: NEXT i
  50 PRINT AT 1,0; INK 5; PAPER
1;"     Press any key to start
 "
  60 IF INKEY$="" THEN  GO TO 60
  70 LET o$="    Press any key t
o start                  'IDIOT'
has trapped you in the back of t
he formroom with desks. You must
 try to shoot Idiot and escape t
hrough the CYAN door.  .   .   .
."
  90 FOR i=1 TO LEN o$-31: PRINT
 AT 1,0; INK 5; PAPER 1;o$(i TO
i+31): FOR n=0 TO 10: NEXT n: BE
EP .001,40: NEXT i
  95 RESTORE 9000: FOR i=144 TO
164: FOR n=0 TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
  97 CLS
 100 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
 110 PRINT AT 0,12; PAPER 2; INK
7;"FORMROOM"
 120 PRINT AT 17,1; INK 6;"BC";
FLASH 1; INK 2; PAPER 7;"EE"; FL
ASH 0; INK 6; PAPER 0;"BC BC BC
BC BC BC BC BC BC"
 130 PRINT AT 18,1; INK 6;"■■";
FLASH 1; INK 2; PAPER 7;"EE"; FL
ASH 0; INK 6; PAPER 0;"■■ ■■ ■■
■■ ■■ ■■ ■■ ■■ ■■"
 140 PRINT AT 1,31; INK 5;"D";AT
 2,31;"E"
 150 PRINT AT 0,2; INK 7; PAPER
0;" HIJKLM "
 151 PRINT AT 21,1; INK 6; PAPER
 1;"Time : 100. Hi :       . By :
 ";a$
 152 PRINT AT 21,18; INK 6; PAPE
R 1;hi
 155 LET x=29: LET a=2: LET b=1:
 LET t=100: LET z=10: LET s=0
 180 PRINT AT 21,1; INK 6; PAPER
```



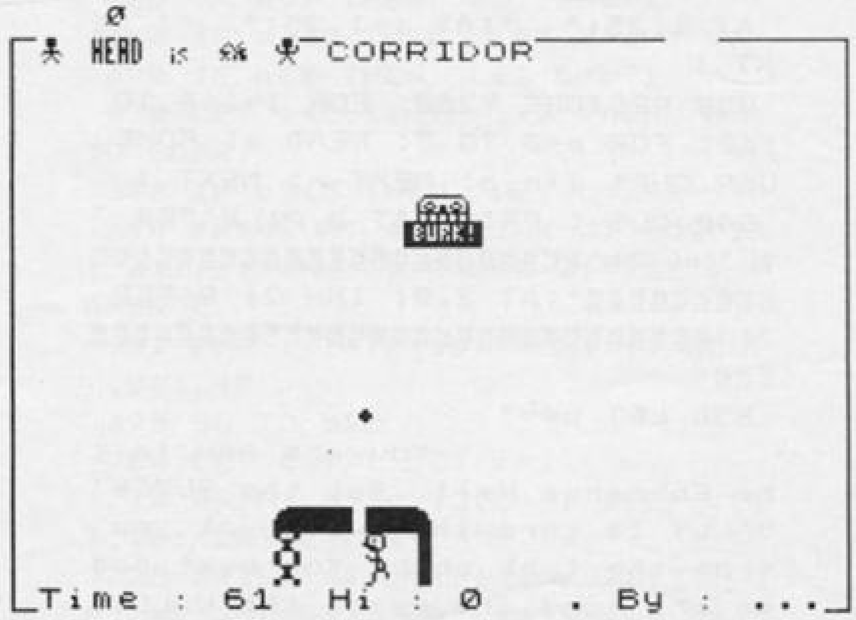

```
 1;"Time : ";t;" "
 190 PRINT AT 19,x;"  ";AT 20,x;
"  "
 200 LET x=x+(INKEY$="8" AND x<2
9)-(INKEY$="5" AND x>1)
 210 PRINT AT 19,x; INK 2;"RS";A
T 20,x; INK 2;"TU"
 220 IF INKEY$="0" THEN  GO TO 3
00
 225 IF z=1 THEN  GO TO 300
 227 PRINT AT a,b;"  ";AT a+1,b;
"  "
 230 IF RND>.5 THEN  LET b=b+1
 240 IF RND<.5 THEN  LET b=b-1
 250 IF RND>.5 THEN  LET a=a+1
 260 IF RND<.5 THEN  LET a=a-1
 265 BEEP .01,-20
 270 IF a<2 THEN  LET a=a+2
 280 IF a>15 THEN  LET a=a-2
 281 LET t=t-1
 282 IF t<0 THEN  GO TO 8000
 290 IF b<2 THEN  LET b=b+2
 291 IF b>29 THEN  LET b=b-2
 292 PRINT AT a,b; INK 3;"NO";AT
 a+1,b; INK 3;"PQ"
 293 GO TO 180
 300 IF z=1 THEN  GO TO 320
 305 IF x=7 OR x=10 OR x=13 OR x
=16 OR x=19 OR x=22 OR x=25 OR x
=28 THEN  GO TO 315
 310 GO TO 225
 315 LET c=18: LET d=x
 320 LET z=1: PRINT AT c,d;" ":
IF c=a AND d=b OR c=a+1 AND d=b
OR c=a AND d=b+1 OR c=a+1 AND d=
b+1 THEN  GO TO 400
 330 LET c=c-1: PRINT AT c,d; IN
K 5;"G": IF c=2 THEN  LET z=10:
PRINT AT 1,d;" ";AT 2,d;" "
 340 GO TO 227
 400 LET s=s+t
 405 FOR i=b TO 3 STEP -1
```

```
410 PRINT AT a,i; FLASH 1; PAPER 7; INK 3;"NO";AT a+1,i; FLASH 1; INK 7; INK 3;"PQ"
415 BEEP .25,i: PRINT AT a,i;"  ";AT a+1,i;"  "
420 NEXT i
430 FOR e=a TO 17: PRINT AT e,3; FLASH 1; PAPER 7; INK 3;"NO";AT e+1,3;"PQ": BEEP .25,e: PRINT AT e,3;"  ";AT e+1,3;"  ": NEXT e
440 FOR i=x TO 3 STEP -1
450 PRINT AT 19,i; INK 2;"RS";AT 20,i; INK 2;"TU"
460 BEEP .02,20: PRINT AT 19,i;"  ";AT 20,i;"  "
470 NEXT i
480 FOR i=17 TO 1 STEP -1: PRINT AT i,3; INK 2;"RS";AT i+1,3;"TU": BEEP .02,21: PRINT AT i,3;"  ";AT i+1,3;"  ": NEXT i
490 FOR i=3 TO 30: PRINT AT 1,i; INK 2;"RS";AT 2,i;"TU": BEEP .02,22: PRINT AT 1,i;"  ";AT 2,i;"  ": NEXT i
495 RESTORE 9100: FOR i=144 TO 160: FOR n=0 TO 7: READ a: POKE USR CHR$ i+n,a: NEXT n: NEXT i
500 CLS
510 LET o$="                                The BURK is lurking in the corridor. He has locked you in your locker. You must fire pellets at him to kill him. The only problem is that you have less time to do it.  .  .  ."
520 PRINT AT 0,0; INK 2; PAPER 7;"GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG"; INK 5; PAPER 1;"                                "; INK 2; PAPER 7;"GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG"
530 FOR i=1 TO LEN o$-31: PRINT AT 1,0; PAPER 1; INK 5;o$(i TO i+31): FOR n=0 TO 5: NEXT n: BEEP .001,40: NEXT i
535 FOR i=1 TO 500: NEXT i
540 CLS : PLOT 0,0: DRAW 255,0: DRAW 0,175: DRAW -255,0: DRAW 0,-175
550 PRINT AT 18,10; INK 6;"J███K";AT 19,10; FLASH 1; PAPER 7; INK 2;"H"; INK 2; FLASH 0; PAPER 0;"  RS"; INK 6;"▌";AT 20,10; INK 2; FLASH 1; PAPER 7;"H"; INK 2; FLASH 0; PAPER 0;"  TU"; INK 6;"▌"
552 PRINT AT 0,1; INK 5;"L"; INK 7;" MN O P "; INK 5;"Q"
555 PRINT AT 21,1; INK 6; PAPER 1;"Time : 75. Hi :      . By : ";a$
560 PRINT AT 21,1; INK 6; PAPER 1;"Time : 75. Hi : ";hi;AT 21,21;". By : ";a$
570 PRINT AT 0,12; PAPER 2; INK 7;"CORRIDOR";AT 0,25; PAPER 0;"      "
580 LET a=2: LET b=10: LET z=10: LET t=75
590 PRINT AT a,b;"   ";AT a+1,b;"   "
595 PRINT AT 21,8; PAPER 1; INK 6;t;" "
600 IF INKEY$="0" THEN GO TO 700
605 IF z=1 THEN GO TO 710
610 IF RND>.5 THEN LET a=a+1
620 IF RND<.5 THEN LET a=a-1
630 IF RND>.5 THEN LET b=b+1
640 IF RND<.5 THEN LET b=b-1
650 IF a<2 THEN LET a=a+1
660 IF a>30 THEN LET a=a-1
670 IF b<2 THEN LET b=b+1
675 LET t=t-1: IF t=0 THEN GO TO 8000
680 IF b>15 THEN LET b=b-1
690 PRINT AT a,b; INK 3;"ABC";AT a+1,b; INK 7; PAPER 2;"DEF": PAPER 0
691 FOR i=1 TO 10: NEXT i: BEEP .001,40
695 GO TO 590
700 LET c=17: LET z=1
710 PRINT AT c,13;" ": BEEP .001,20: LET c=c-1
720 PRINT AT c,13; INK 7;"I": IF c=a AND b=13 OR c=a+1 AND b=13
```



```
OR c=a AND b=12 OR c=a+1 AND b=
12 OR c=a AND b=11 OR c=a+1 AND
b=11 THEN  GO TO 75Ø
 725 IF c=1 THEN  LET z=1Ø: PRIN
T AT c,13;" "
 73Ø GO TO 61Ø
 75Ø FOR i=b TO 5 STEP -1: PRINT
 AT a,i; INK 7; FLASH 1; PAPER 3
;"ABC";AT a+1,i; INK 7; PAPER 2;
"DEF": BEEP .25,i: PRINT AT a,i;
"   ";AT a+1,i;"   ": NEXT i
 755 LET s=s+t
 76Ø FOR i=a TO 19: PRINT AT i,5
; INK 7; FLASH 1; PAPER 3;"ABC";
AT i+1,5; INK 7; PAPER 2;"DEF":
BEEP .25,i: PRINT AT i,5;"   ";A
T i+1,5;"   ": NEXT i
 77Ø FOR i=5 TO 1Ø: PRINT AT 19,
i; INK 7; PAPER 3; FLASH 1;"ABC"
;AT 2Ø,i; INK 7; PAPER 2;"DEF":
BEEP .25,i: PRINT AT 19,i;"   ";
AT 2Ø,i;"   ": NEXT i
 78Ø FOR i=1 TO 16: PRINT AT i,1
Ø; PAPER Ø; FLASH Ø;"          ": N
EXT i
 79Ø FOR i=13 TO 3 STEP -1: PRIN
T AT 19,i; INK 2;"RS";AT 2Ø,i; I
NK 2;"TU": BEEP .Ø12,2Ø: PRINT A
T 19,i;"  ";AT 2Ø,i;"  ": NEXT i
 8ØØ FOR i=19 TO 1Ø STEP -1: PRI
NT AT i,3; INK 2;"RS";AT i+1,3;
INK 2;"TU": BEEP .Ø12,21: PRINT
AT i,3;"  ";AT i+1,3;"  ": NEXT
i
 81Ø FOR i=3 TO 25: PRINT AT 1Ø,
i; INK 2;"RS";AT 11,i; INK 2;"TU
": BEEP .Ø12,22: PRINT AT 1Ø,i;"
 ";AT 11,i;"  ": NEXT i
 82Ø FOR i=1Ø TO Ø STEP -1: PRIN
T AT i,25; INK 2;"RS";AT i+1,25;
 INK 2;"TU": BEEP .Ø12,23: PRINT
 AT i,25;"  ";AT i+1,25;"  ": NE
XT i
 83Ø RESTORE 92ØØ: FOR i=144 TO
16Ø: FOR n=Ø TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
 84Ø CLS : PRINT AT Ø,Ø; PAPER 7
; INK 2;"EEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEE";AT 2,Ø; INK 2; PAPER
7;"EEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEE"
 85Ø LET o$="
                  You are now in t
he Entrance Hall. But the SUPER
WALLY is throwing sweets at you
from the tuck shop. You must dod
ge the food and reach the WALLY
ALARM BELL. Then the nearest tea
cher will suspend all the WALLY'
S!!!   .   .   .   .   ."
 86Ø FOR i=1 TO LEN o$-31: PRINT
 AT 1,Ø; PAPER 1; INK 5;o$(i TO
i+31): BEEP .ØØ1,4Ø: FOR j=Ø TO
5: NEXT j: NEXT i
 87Ø FOR i=1 TO 5ØØ: NEXT i
 88Ø CLS : PLOT Ø,Ø: DRAW 255,Ø:
 DRAW Ø,175: DRAW -255,Ø: DRAW Ø
,-175: PRINT AT Ø,9; INK 7; PAPE
R 2;"ENTRANCE HALL.";AT Ø,2; PAP
ER Ø;"      "
 89Ø PRINT AT 21,1; INK 6; PAPER
 1;"
  ";AT 21,1; INK 6; PAPER 1;"Tim
e : 5Ø. Hi : ";hi;AT 21,21;". By
 : ";a$
 9ØØ PRINT AT 1Ø,29; INK 6;"EO■"
;AT 11,29; INK 6;" "; INK 3;"AB"
;AT 12,29; INK 6;"■"; INK 3;"CD"
;AT 13,29; INK 6;"GH▄"
 91Ø LET o$="I JK L I J KK LJ KL
 I JK L JK L  JK K L JK K J  KJ
L J L JK JLK KJ KJLI JKLJI"
 915 LET y=2: LET x=2: PRINT AT
2Ø,1Ø; INK 5;"MN": LET t=5Ø
 92Ø FOR i=1 TO LEN o$-28: PRINT
 AT 11,1; INK 7; PAPER Ø;o$(i TO
 i+28): BEEP .ØØ1,4Ø
 93Ø PRINT AT y,x;"  ";AT y+1,x;
"  "
 94Ø LET y=y+(INKEY$="6" AND y<1
9)-(INKEY$="7" AND y>2)
 943 LET t=t-1: IF t=Ø THEN  GO
TO 8ØØØ
 944 PRINT AT 21,8; INK 6; PAPER
 1;t;" ": PAPER Ø
 945 LET x=x+(INKEY$="8" AND x<2
5)-(INKEY$="5" AND x>2)
 946 IF SCREEN$ (y,x)<>" " AND S
CREEN$ (y+1,x)<>" " AND SCREEN$
(y,x+1)<>" " AND SCREEN$ (y+1,x+
1)<>" " THEN  GO TO 799Ø
 95Ø PRINT AT y,x; INK 2;"RS";AT
y+1,x;"TU"
 96Ø NEXT i
 97Ø GO TO 92Ø
1ØØØ FOR i=1 TO 6: BEEP .Ø1,2Ø:
BEEP .Ø1,21: BEEP .Ø1,22: BEEP .
Ø1,23: BEEP .Ø1,22: BEEP .Ø1,21:
 NEXT i
1ØØ1 LET s=s+t
1Ø1Ø FOR i=Ø TO 5: PRINT AT i,3;
 INK 2;"PQ";AT i+1,3;"TU": BEEP
.Ø12,2Ø: PRINT AT i,3;"  ";AT i+
1,3;"  ": NEXT i
```

```
1020 FOR i=3 TO 30: PRINT AT 5,i
; INK 2;"PQ";AT 6,i;"TU": BEEP .
012,21: PRINT AT 5,i;"  ";AT 6,i
;"  ": NEXT i
1030 FOR i=5 TO 10: PRINT AT i,3
0; INK 2;"PQ";AT i+1,30;"TU": BE
EP .012,22: PRINT AT i,30;"  ";A
T i+1,30;"  ": NEXT i
1040 FOR i=1 TO 50: NEXT i
1050 FOR i=10 TO 5 STEP -1: PRIN
T AT i,30; INK 2;"PQ";AT i+1,30;
"TU";AT i+2,30; INK 3;"AB";AT i+
3,30; INK 3;"CD": BEEP .012,23:
PRINT AT i,30;"  ";AT i+1,30;"
";AT i+2,30;"  ";AT i+3,30;"  ":
 NEXT i
1060 FOR i=30 TO 3 STEP -1: PRIN
T AT 5,i; INK 2;"PQ";AT 6,i;"TU"
;AT 7,i; INK 3;"AB";AT 8,i; INK
3;"CD": BEEP .012,24: PRINT AT 5
,i;"  ";AT 6,i;"  ";AT 7,i;"  ";
AT 8,i;"  ": NEXT i
1070 FOR i=5 TO 0 STEP -1: PRINT
 AT i,3; INK 2;"PQ";AT i+1,3;"TU
";AT i+2,3; INK 3;"AB";AT i+3,3;
 INK 3;"CD": BEEP .012,25: PRINT
 AT i,3;"  ";AT i+1,3;"  ";AT i+
2,3;"  ";AT i+3,3;"  ": NEXT i
1080 RESTORE 9070: FOR i=148 TO
151: FOR n=0 TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
1090 RESTORE 9100: FOR i=152 TO
157: FOR n=0 TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
1100 RESTORE 9300: FOR i=158 TO
160: FOR n=0 TO 7: READ a: POKE
USR CHR$ i+n,a: NEXT n: NEXT i
1101 CLS : PRINT AT 21,7; PAPER
2; INK 7;"HEADMASTER'S STUDY."
1102 PAPER 0
1110 PRINT AT 15,10; INK 3;"EF
 IJK  AB";AT 16,10; INK 3;"GH
"; INK 7; PAPER 2;"LMN"; INK 3;
 PAPER 0;"  CD"
1120 PRINT AT 2,0; INK 7;"Ha Ha!
 "; INK 2;"RS";AT 3,7; INK 2;"TU
"
1130 FOR i=1 TO 100: NEXT i
1140 FOR i=1 TO 15: PRINT AT i,1
0; INK 7;"Q";AT i+1,10; INK 7;"P
Q": FOR n=0 TO 10: NEXT n: PRINT
 AT i+1,10;"  ";AT i,10;" ": BEE
P .009,i: NEXT i
1150 BEEP 1,-40
1160 FOR i=1 TO 15: PRINT AT i,1
5; INK 7;"Q";AT i+1,15; INK 7;"P
Q": FOR n=0 TO 10: NEXT n: PRINT
 AT i+1,15;"  ";AT i,15;"  ":
BEEP .009,i: NEXT i
1170 BEEP 1,-40
1180 FOR i=1 TO 15: PRINT AT i,2
1; INK 7;"Q";AT i+1,21; INK 7;"P
Q": FOR n=0 TO 10: NEXT n: PRINT
 AT i+1,21;"  ";AT i,21;" ": BEE
P .009,i: NEXT i
1190 BEEP 1,-40
1195 PRINT AT 0,0;
1200 RESTORE 9400: FOR i=1 TO 8:
 READ a: READ b: BEEP a/4,b: NEX
T i
1201 LET c=1
1205 FOR i=0 TO 6
1210 PRINT  INK c;"█▌█ ▛▀▘ █  █
  ▛▖▞▛▖▛▜▗▛▀"
1220 PRINT  INK c+1;"█▌█ ▛▀ █
 █  █ ██ ██ ██ █▛▀"
1230 PRINT  INK c+2;"▀▀▀ ▀▀▀ ▀▀
 ▀▀▀ ▀▀ ▀▀ ▀ ▀ ▀▀"
1240 LET c=c+3: IF c>7 THEN  LET
 c=1
1250 NEXT i
1260 FOR i=1 TO 100: NEXT i
1270 RESTORE 9000: FOR n=0 TO 7:
 READ a: POKE USR CHR$ 144+n,a:
NEXT n: NEXT i
1275 CLS
1280 GO TO 8020
7989 STOP
7990 IF x=10 AND y=19 OR x=9 AND
 y=19 THEN  GO TO 1000
8000 FOR i=21 TO 0 STEP -1: PRIN
T AT i,0; PAPER 0;"
                   ": BEEP .12,i
: NEXT i
8005 RESTORE 9000: FOR n=0 TO 7:
 READ a: POKE USR CHR$ 144+n,a:
NEXT n: NEXT i
8010 BEEP 1,0: BEEP 1,5: BEEP 1,
0
```



```
8020 PRINT AT 0,0; INK 3; PAPER
7;"Score : ";s;TAB 12;". Hi : ";
hi;TAB 23;".By : ";a$
8030 IF s>hi THEN  GO TO 8100
8040 PRINT AT 2,0; INK 6; PAPER
2;"Unfortunatly ,you didn't get
the              High-score
  "
8045 FOR i=1 TO 500: NEXT i
8050 PRINT AT 21,0; INK 2; PAPER
 7;"AAPRESS ANY KEY TO PLAY AGAI
N.AA"
8060 IF INKEY$="" THEN  GO TO 80
60
8065 CLS
8067 LET s=0
8070 LET o$="
             'HUDDLESTON' has tra
pped you in the back of the form
room with desks. You must try to
 shoot Huddleston and escape thr
ough the CYAN door.  .  .  .  ."
8075 PRINT AT 0,0; INK 2; PAPER
7;"AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA";AT 2,0; INK 2; PAPER 7;"AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
8080 GO TO 90
8100 PRINT AT 2,0; PAPER 7; INK
1;"    YOU HAVE THE HIGH-SCORE!
  "
8110 INPUT  PAPER 2; INK 7;"PLEA
SE ENTER YOUR INITIALS...... (m
ax. 3 letters) : "; LINE a$
8120 IF LEN a$<>3 THEN  GO TO 81
10
8122 LET hi=s
8125 PRINT AT 0,0; INK 3; PAPER
7;"Score : ";s;TAB 12;". Hi : ";
hi;TAB 23;".By : ";a$
8130 GO TO 8045
9000 DATA 192,48,12,3,3,12,48,19
2
9010 DATA 255,254,254,255,128,25
5,255,255,255,127,127,255,1,255,
255,255
9013 DATA 15,123,227,227,227,239
,243,227,227,243,239,227,163,227
,123,15
9016 DATA 195,195,60,24,24,60,19
5,195
9030 DATA 16,56,116,238,91,54,28
,8
9040 DATA 255,130,130,124,68,68,
40,40
9050 DATA 252,130,130,68,72,80,9
6,64
9060 DATA 255,136,136,73,73,73,4
2,42,160,160,160,31,17,17,10,10,
128,128,128,2,5,7,5,21,0,0,0,55,
68,71,68,55
9070 DATA 15,16,161,225,99,110,1
10,99,240,8,133,135,198,118,118,
198,161,33,40,47,47,39,16,15,133
,132,20,244,244,228,8,240
9080 DATA 7,8,10,24,14,7,1,9,128
,64,64,64,64,128,0,0,7,1,1,1,2,2
,2,12,128,64,64,192,32,32,0,0
9100 DATA 3,4,9,10,9,15,9,9,255,
0,129,90,153,255,36,36,192,32,14
4,80,144,240,144,144
9110 DATA 255,230,234,230,234,23
0,255,255,255,162,170,166,170,42
,255,255,255,183,119,247,127,183
,255,255
9120 DATA 192,48,12,3,3,12,48,19
2
9130 DATA 195,195,60,24,24,60,19
5,195
9140 DATA 0,0,24,60,60,24,0,0
9150 DATA 15,63,127,127,255,255,
255,255,240,252,254,254,255,255,
255,255
9160 DATA 28,28,28,8,62,73,20,34
,87,84,84,118,118,84,84,87,118,8
5,85,117,117,85,85,86,0,0,0,76,1
6,72,68,80,0,34,84,139,230,75,17
1,0
9170 DATA 28,28,93,73,62,8,20,34
9200 DATA 15,16,166,233,102,96,1
05,239,240,8,101,151,102,6,150,2
47,175,47,47,47,47,39,16,15,245,
244,244,244,244,228,8,240
9210 DATA 192,48,12,3,3,12,48,19
2
9220 DATA 15,63,127,127,255,255,
255,255,255,255,255,255,127,127,
63,15
9230 DATA 192,240,248,252,255,25
5,255,255
9240 DATA 4,6,31,60,60,248,96,32
,0,24,60,102,102,60,24,0,0,8,28,
28,62,62,28,0
9250 DATA 8,28,58,119,46,28,8,0
9260 DATA 99,111,159,191,129,255
,255,255,192,240,248,252,128,255
,255,255
9270 DATA 255,255,255,255,252,24
8,240,192,0,7,67,134,76,39,19,9,
32,192,128,64,64,64,128,0
9300 DATA 0,127,130,66,65,34,68,
130,65,130,130,129,128,128,134,1
21,0,0,211,47,1,66,180,8
9400 DATA 1,20,0.5,20,0.5,20,1,2
0,1,24,1,20,1,24,2,27
```

Twenty Ones (or, Dicey-Dicey)

Timothy Simpson lives in Bradford and has sent this infuriating game to share his frustration around.

Full instructions are included in the game which is based on the card game Pontoon. Four dice are thrown and the player then has the option to Twist — throw another dice, or Stick.

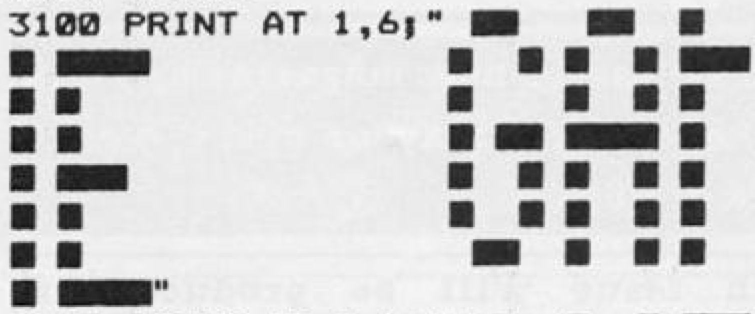
The aim of the game is to get a total closer to twenty one than the computer without going over — going bust. Should you or the computer go bust then the opponent wins, also should you make twenty one then you automatically win. You have an advantage because you always go first.

OK. Roll up your sleeves, breathe on the keyboard for luck, and thrown them bones.

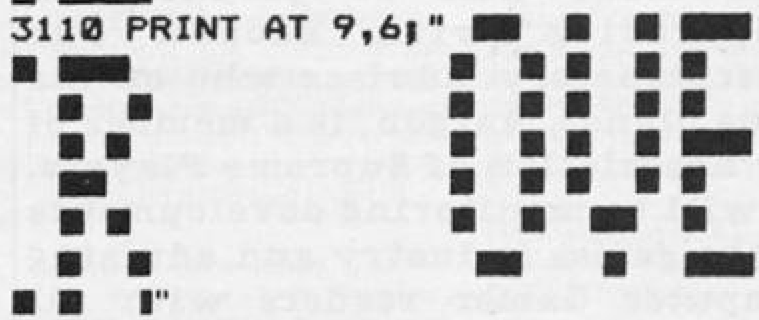
```
  10 PRINT AT 0,5;"****DICEY*DIC
EY****"
  11 PRINT AT 3,0;"THIS GAME IS
SIMILAR TO PONTOON.DICEY DICEY I
S PLAYED WITH FOUR DICE."
  13 PRINT AT 10,0;"PRESS ANY KE
Y PLEASE"
  14 GOSUB 1000
  16 PRINT AT 2,0;"I WILL ROLL Y
OUR DICE WHICH WILLBE DISPLAYED
GRAPHICALLY."
  17 PRINT AT 6,0;"A TOTAL WILL
ALSO BE DISPLAYED, WHICH IS IMPO
RTANT.
                      THE OBJECT OF
 THE GAME IS TO GETTHE TOTAL AS
NEAR TO,OR EQUAL TO21 AS POSSIBL
E."
  19 PRINT AT 16,0;"PRESS ANY KE
Y"
  20 GOSUB 1000
  40 PRINT AT 3,0;"PRESS ANY KEY
 TO ROLL YOUR DICE"
  41 GOSUB 1000
  42 PRINT AT 0,0;"DICE(1 TO 4)
                 ▀▀▀▀▀▀▀▀▀▀"
  43 PRINT AT 3,0;"▛▀▜ ▛▀▜ ▛▀▜ ▛
▀▜"
  44 PRINT AT 4,0;"▌ ▐▌ ▐▌ ▐▌ ▐
▌"
  45 PRINT AT 5,0;"▙▄▟ ▙▄▟ ▙▄▟ ▙
▄▟"
  46 LET D1=INT (RND*6)+1
  47 PRINT AT 4,1;D1
  48 LET D2=INT (RND*6)+1
  49 PRINT AT 4,5;D2
  50 LET D3=INT (RND*6)+1
  51 PRINT AT 4,9;D3
  52 LET D4=INT (RND*6)+1
  53 PRINT AT 4,13;D4
  54 LET TWIST=0
  55 PRINT AT 10,0;"TOTAL:";(D1+
D2+D3+D4)+TWIST
  56 IF TWIST+(D1+D2+D3+D4)>21 T
HEN GOTO 400
  57 IF TWIST+(D1+D2+D3+D4)=21 T
HEN GOTO 300
  58 LET D1=D1+TWIST
  60 PRINT AT 11,0;"************
********"
  62 PRINT AT 12,0;"(S)TICK OR (
T)WIST ?"
  63 LET A$=INKEY$
  64 IF A$="S" THEN PRINT AT 16,
0;"YOUR TOTAL IS NOW ENTERED"
  65 IF A$="S" THEN GOTO 68
  66 IF A$="T" THEN GOTO 80
  67 IF A$<>"S" AND A$<>"T" THEN
 GOTO 60
  68 GOTO 90
  80 PRINT AT 16,0;"ANOTHER DI I
S NOW THROWN"
  82 LET TWIST=INT (RND*6)+1
  83 FOR A=1 TO 30
  84 NEXT A
  85 PRINT AT 8,0;"";TWIST
  86 PRINT AT 16,0;"
              "
  88 GOTO 55
  90 GOSUB 3000
 100 PRINT AT 0,0;"MY TURN"
 105 FOR A=1 TO 20
 106 NEXT A
 107 CLS
 110 PRINT "DICE(1 TO 4)
                 ▀▀▀▀▀▀▀▀▀▀"
 111 PRINT AT 3,0;"▛▀▜ ▛▀▜ ▛▀▜ ▛
▀▜"
 112 PRINT AT 4,0;"▌ ▐▌ ▐▌ ▐▌ ▐
▌"
 114 PRINT AT 5,0;"▙▄▟ ▙▄▟ ▙▄▟ ▙
▄▟"
 117 LET D5=INT (RND*6)+1
 118 PRINT AT 4,1;D5
 120 LET D6=INT (RND*6)+1
 122 PRINT AT 4,5;D6
 124 LET D7=INT (RND*6)+1
 126 PRINT AT 4,9;D7
 128 LET D8=INT (RND*6)+1
```

```
130 PRINT AT 4,13;D8
140 PRINT AT 10,0;"TOTAL:";D5+D
6+D7+D8
142 PRINT AT 11,0;"************
********"
144 LET C=(D5+D6+D7+D8)
145 LET H=(D1+D2+D3+D4)
146 IF C<=17 THEN GOTO 155
147 IF C>17 AND C<21 THEN GOTO
2000
148 IF C>21 THEN GOTO 500
150 IF C=21 THEN GOTO 200
152 IF H<C THEN GOTO 200
153 IF C<H THEN GOTO 300
154 IF C=H THEN GOTO 700
155 PRINT AT 14,0;"I WILL TWIST
"
156 LET TWIST=INT (RND*6)+1
157 FOR A=1 TO 25
158 NEXT A
159 PRINT AT 8,0;"";TWIST
160 PRINT AT 14,0;"
"
162 PRINT AT 10,0;"TOTAL:";(D5+
D6+D7+D8)+TWIST
164 LET D5=D5+TWIST
165 GOTO 144
200 GOSUB 3000
201 GOSUB 3100
207 PRINT AT 10,13;"I WIN"
208 FOR A=1 TO 45
209 NEXT A
210 GOSUB 3000
220 PRINT AT 0,0;"PRESS ANY KEY
TO PLAY AGAIN"
230 IF INKEY$="" THEN GOTO 230
240 GOSUB 3000
250 GOTO 38
300 GOSUB 3000
301 GOSUB 3100
306 PRINT AT 10,12;"YOU WIN"
307 FOR A=1 TO 45
308 PAUSE 50
310 GOSUB 3000
320 PRINT AT 0,0;"PRESS ANY KEY
TO PLAY AGAIN"
330 IF INKEY$="" THEN GOTO 330
345 GOSUB 3000
350 GOTO 38
400 GOSUB 3000
401 PRINT AT 0,0;"YOU";
402 GOSUB 3200
404 GOTO 201
405 PRINT AT 5,0;"PRESS ANY KEY
TO PLAY AGAIN"
410 IF INKEY$="" THEN GOTO 410
420 GOSUB 3000
430 GOTO 38
450 GOSUB 3000
455 GOTO 200
500 GOSUB 3000
501 PRINT AT 0,0;"I";
502 GOSUB 3200
510 PRINT AT 0,0;"I HAVE BUST"
515 FOR A=1 TO 35
516 NEXT A
520 GOTO 300
700 GOSUB 3000
710 PRINT AT 0,0;"THAT GAME WAS
A DRAW"
711 PAUSE 100
715 GOSUB 3000
720 GOTO 405
1000 IF INKEY$="" THEN GOTO 1000

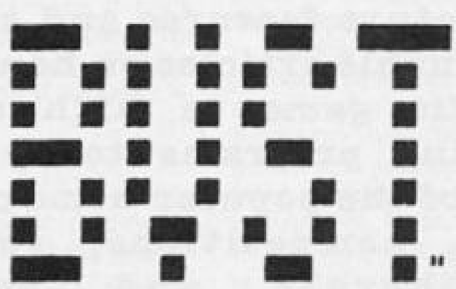
1015 GOSUB 3000
1020 RETURN
2000 FOR A=1 TO 20
2010 NEXT A
2020 GOTO 148
3000 FOR A=1 TO 21
3010 SCROLL
3011 NEXT A
3020 RETURN
3100 PRINT AT 1,6;"
```



```
"
3110 PRINT AT 9,6;"
```



```
"
3120 PAUSE 50
3130 GOTO 3000
3200 PRINT " HAVE
```



```
"
3210 RETURN
4000 SAVE "DICEY-DICE"
4010 RUN
```



De-bugger

Getting a program typed in is often only the start of your problems. Ed to the rescue.



Typing in a program is a useful exercise. Apart from the patience required, techniques learned and the end program to be used, probably the most educational part of it is tracking down the bugs introduced by yourself or occasionally by our publication system.

In debugging you gain a much deeper insight and understanding on how the program actually works than by merely typing it in, but tracking down these errors is an art in itself and needs some skill. So here are some tips to help you in your efforts when faced with that cryptic error report!

1 NEXT without FOR
Look back through the program, either the loop has not been set up — no related FOR 'letter' = No1 TO No2 line, or the letter has been re-used as an ordinary variable within the loop with a LET 'letter' = No.

2 Variable not found
This is one of the most common errors. Again, the problem may not lie in the line where the error was detected and reported. If there is only one variable, which may be one or more letters or a string ($) variable, then that is the problem. There may be more than one variable in the line section reported and you will have to identify the offending one. In a line PRINT AT Y,X;A$ the culprit could be Y or X or A$. To find out which of them is causing the problem (it may be more than one) type in turn as a direct command:

PRINT Y Enter/Newline
PRINT X Enter/Newline
PRINT A$ Enter/Newline

Note which produces the error report. Now look back through the program printout for the line which sets it up — usually a LET or FOR command. Did you leave it out? Does the program get there or has a GOTO/GOSUB been wrongly addressed?

3 Subscript wrong
Connected with DIM A(No) or DIM A$(No). If the number in the brackets on the line where the error is reported is greater than the one in the original DIM statement, is not an integer or is less than 1, then this report is generated. If the subscript — number in brackets — is a number then check and change, however, if it is a variable then follow the procedure for tracing variables. It has probably exceeded the limits, look for lines with the variable being altered with + − * / : if necessary add limiting code. For example:

IF X >10 THEN LET X = 10

4 Out of memory
As well as for programs which are too big, it may happen if the previous program set RAMtop. Before despairing, enter CLEAR USR "a"-1 on the Spectrum: on the ZX81 SAVE the program, turn the machine off and on, then reload the program.

7 RETURN without GOSUB
Somehow the computer has reached a RETURN command other than via a GOSUB instruction. Check a GOTO hasn't been entered in place of a GOSUB. Check for a missing GOSUB.

B Integer out of range
An integer (whole number) either as a number of variable is too big or small and you are attempting to do something like PRINT AT 0,33 — not allowed! Check any variables involved as per report 2 and trace it back through the program looking for adjustments to it by + − * / : Add limiting code if needed — see report 3

E Out of DATA
A Spectrum problem. Check the number of DATA items match the number of READs; usually one (or more) has been missed out. Attempting to reread a DATA list without first using a RESTORE command will cause this and it can happen on an auto start program (saved with a LINE number). Good programming usually RESTOREs to the correct line number before using READ.

I FOR without NEXT
See report 1 but this time the NEXT is missing!

Note that the letters I have used for examples could by ANY letters not just A$,X,Y etc and depend on the particular choice of the programmer.

This is by no means a comprehensive list but I have tried to cover many of the most common error reports. Personally, I get almost as much satisfaction from debugging as I do from programming I do assure you, however, that there is absolutely no truth in the rumour that we deliberately inject bugs into our listings in order to introduce you to the dubious delights of debugging! ■

Conversion tips

A guide to ZX81/Spectrum program conversions from David Nowotnik.

The versions of BASIC offered by the two ZX computers are so similar that many programs for one can be used by the other. The ZX81 has only two commands which are not present on the Spectrum, SCROLL and UNPLOT, and these should cause you few problems when converting ZX81 programs to the Spectrum (see Table 1).

There are quite a lot of commands and functions on the Spectrum which are not available on the ZX81. A list of these appears in Table 4. The stars indicate those commands and functions for which there is no simple translation to ZX81 BASIC. Those for colour and sound can be ommitted; but you will have to find some alternative for the high resolution and file I/O commands.

The command PLOT appears on both computers, but the effect is quite different, so beware! Another tip: PEEK and POKE should be used with caution. In conversion, addresses will almost certainly have to be changed. Some of those changes appear in the tables. A command such as POKE USR "a".... on the Spectrum indicates User Defined Graphics; ZX81 users don't have this facility, so you'll have to omit this and use a standard character instead.

ZX81	Spectrum	Comments
SCROLL	RANDOMISE USR 3582 or LET t=USR 3582	If the program uses random numbers, they could become rather predictable with the first option. If so, use the second, using a variable (in this case t) which is otherwise not used.
PLOT Y,X	PRINT AT 21−Y/2,X/2;	Print the appropriate quarter square graphics character.
UNPLOT Y,X	PRINT AT 21−Y/2,X/2;	Print a space, or the appropriate quarter square graphics character.

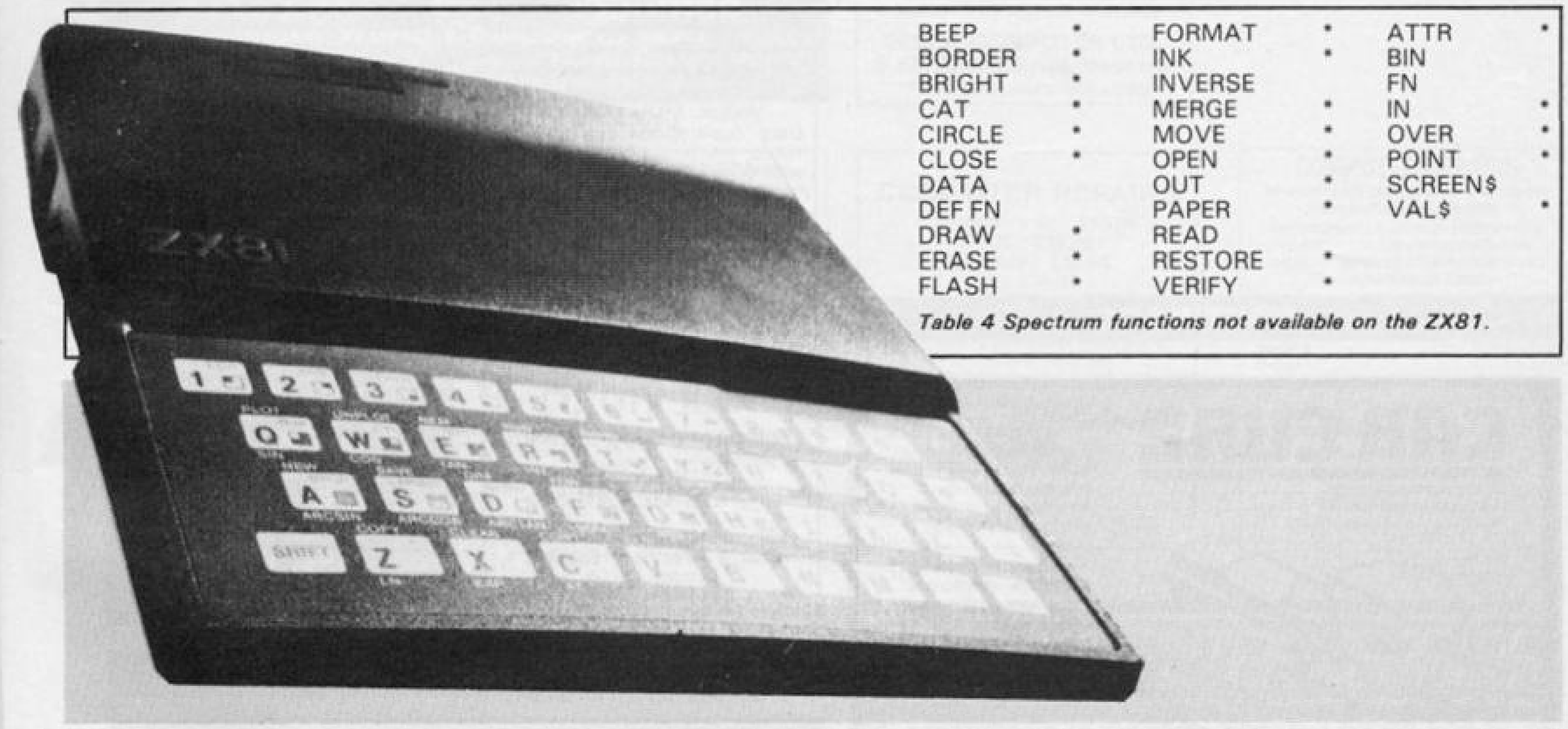
Table 1 ZX81 to Spectrum conversions.

Spectrum	ZX81	Comments
BIN eg LET y=BIN 10010101	LET y=(decimal no.) Conversion to decimal: 10010101=149 128 64 32 16 8 4 2 1 Add these numbers together when a 1 appears at the appropriate position in binary.	BIN allows the representation of a number in binary. On the ZX81 use the decimal equivalent, but beware; BIN is often used with User Defined Graphics, which are not available on the ZX81.
READ/DATA eg READ x,y DATA 50,60	LET LET X=50 LET Y=60	READ and DATA are used to store a lot of information in a program. Use LET instead.
DEF FN and FN eg DEF a(x)=SQR x LET t=FN a(i)	LET X$="SQR X" LET X=I LET T= VAL X$	The defined function can appear in a string. Use the keyword for built-in functions (eg SQR). The equivalent of FN may need 2 lines, as shown.
PLOT	no equivalent	
SCREEN$ eg LET a=SCREEN$ x,y	LET A=PEEK(PEEK 16396 +256*PEEK 16397+1+Y+33*X)	Used in interactive games to detect characters in the display file. Note — this formula only works when a RAM pack is fitted.

Table 2 Spectrum to ZX81 conversions.

ZX81	Spectrum	Comments
1 FRAMES POKE 16436,255 POKE 16437,255 LET T=(65535−PEEK 16436−256*PEEK 16437) /50	POKE 23672,0:POKE 23673,0 LET t=(PEEK 23672+256* PEEK 23673)/50 For times greater than 10 minutes, you can use byte 23674 as well.	Both computers have a counter which accurately varies by 50 every second. In the example, use the first line to start the 'clock'. The variable T will have the time in seconds after the start. The counter can only be used for 10 minutes.
2 Line number zero POKE 16510,0	POKE 23756,0 (As the start of BASIC can move, eg with microdrives) use with caution.	Converts the first line of a program to line number zero. which cannot be edited, and so is protected.
3 RAMTOP POKE 16388,X−256*INT (X/256) POKE 16389, INT (X/256)	CLEAR x	Creates a safe area at the top of RAM starting at address x, for storing data, machine code etc.

Table 3 General interconversion hints.



BEEP	*	FORMAT	*	ATTR	*
BORDER	*	INK	*	BIN	
BRIGHT	*	INVERSE		FN	
CAT	*	MERGE	*	IN	*
CIRCLE	*	MOVE	*	OVER	*
CLOSE	*	OPEN	*	POINT	*
DATA		OUT	*	SCREEN$	
DEF FN		PAPER	*	VAL$	*
DRAW	*	READ			
ERASE	*	RESTORE	*		
FLASH	*	VERIFY	*		

Table 4 Spectrum functions not available on the ZX81.

System Variables Conversion Table.

Variable	ZX81/ T/S1000	Spectrum/ TS2068
BREG	16414	23655
CDFLAG	16443	No Equivalent
CH ADD	16406	23645
COORDS	16438	23677
COORDS (Byte 2)	16439	23678
DEST	16402	23629
DF CC	16398	23684
D FILE	16396	No Equivalent
DF SZ	16418	23659
E LINE	16404	23641
ERR NR	16384	23610
E PPC	16294	23625
ERR SP	16386	23613
FLAGS	16385	23611
FLAGX	16429	23665
FRAMES	16436	23672
LAST K	16421	23560
MARGIN	16424	No Equivalent
MEM	16415	23656
MEMBOTT	16477	23698
MODE	16390	23617
NXTLIN	16425	23637
OLDPCC	16427	23662
PPC	16391	23621
PRBUFF	16444	23296
PR CC	16440	23680
RAMTOP	16388	23730
SEED	16434	23670
S PSN	16441	23688
S POSN (Byte 2)	16442	23689
STKBOT	16410	23651
STKEND	16412	23653
S TOP	16419	23660
STRLEN	16430	23666
T-ADDR	16432	23668
VARS	16400	23627
VERSN	16393	No Equivalent
X PTR	16408	23647

ZX COMPUTING

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
DAVID HORNE
ASP Ltd., 1 Golden Square,
London W1

ACCESSORIES

SPECTRUM
RGB
OUTPUT
MODULE

adapt electronics

- Connect a monitor to your Spectrum
- Super sharp text. Superb colour graphics •
- True TTL/Analogue outputs + comp syncs
- Monitor quality pictures – no dot crawl
- Suits RGB monitors & monitor/TVs
- Simply plugs into expansion port
- No separate power supply required
- New through port version. Allows connection of modems, joy-stick interfaces, etc

Price £34.95 or £36.95 (through port) + £1 p&p many addons part & software SAE for details

20 Starling Close, Buckhurst Hill, Essex IG9 5TN
Tel. 01-504 2840

LIBRARIES

Free Spectrum Software loan when you join our library. Games educational/utilities available. Full details from 'Cygnet Software' 412 Fulbridge Road, Werrington, Peterborough PE4 6SE.

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

SERVICES

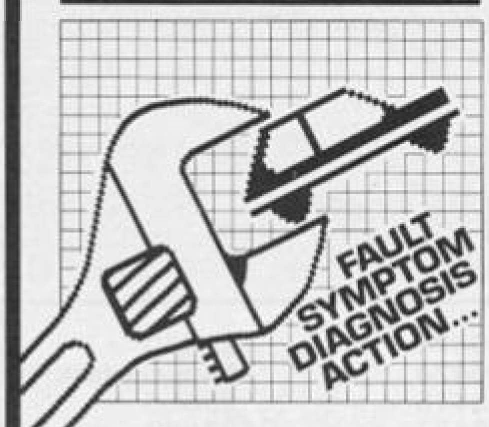
DATA DUPLICATION
MG. Copies (ZX), Burntwood, Walsall, Staffs WS7 0ES.
Data duplication, quality cassettes, printing, flexible delivery with competitive prices and fast turn around. Prices and samples on request. **0543-480887** or **05436-75375** (24 hrs)

FOR SALE

PHILLIPS COMPUTER monitor plus 3 games for the ZX Spectrum: Combat Lynx, Hobbit, Winged Warlords Plus many other similar games. £70.00 ONO. Tel; 0306 880949.

SOFTWARE

CAR CURE



★ **REVIEWS**
"... works extremely well, a great deal of effort has gone into making it user friendly as well as functional ... a very worthwhile package" *-CRASH*
"... an expert system ... incredibly easy to use" *-Popular Computing Weekly*
"... well produced ... a worthwhile addition to your toolkit" *-ZX Computing*
"... one of the most useful programs I have ever used. Makes 'xyz' look very childish and useless. A software company to keep an eye on" *-ZX Appeal (Canada)*

★ **COMPREHENSIVE**
+ 900 faults
+ 300 symptoms

★ **POWERFUL**
+ Discover faulty parts for every symptom
+ Display faults for every part
+ Recommended cure for every fault
+ Trace symptoms of any fault
+ Identify faults causing two symptoms
+ Mark items for easy reference
+ Eliminate faults checked OK
+ Review progress on summary screen
+ Call help if in difficulty
+ Save diagnosis in data file

CAR CURE will transform your micro into an **expert mechanic** to help you **find faults** on your car and then **determine the cure.**
Technical Colleges buy **CAR CURE** to help students **learn** about the faults cars suffer, as well as their symptoms and cures. Now you too can benefit!
Complete with comprehensive **5000-word User Guide** and packaged in a long-life plastic wallet. Export orders get American-English cross reference. All for less than many garages charge for an hour's work!

Cassette for **48K Spectrum** *(microdrive transfer instructions free, on request)* and **Amstrad CPC464** at only £9.95.
Disc for **BBCB** *(state 40 or 80 track)* at £11.95.

SIMTRON
PROGRAMS TO HELP YOU
DEPT ZX, 4 CLARENCE DRIVE, EAST GRINSTEAD, WEST SUSSEX RH19 4RZ **Telephone (0342) 28188**

VIDEO

XXX VIDEO
Telephone (day or night)
0373 - 830563
FOR DETAILS
Glossy list from: **XXX Video Club, 2 Streatham High Rd., London SW16**
(You must be over 18 to ask for details)

SOFTWARE

CONFUZION
BY
INCENTIVE
£6.95

UTILITIES

TAPE TO DRIVE TRANSFER OF RECENT PROGRAMS

NEW

SPECTRUM owners can even transfer many of the **LATEST** *fast/jerky programs to their* drives at **LOW COST** with our software and information sheets. No risk as our programs carry our **MONEY BACK GUARANTEE** (not updates).

Firstly you will need our MD1b (for m/drive) or WD1b (for Wafa or disc drive) programs that give you the tools to convert programs. Includes the ability to chop/split bytes in one go, bytes mover, "VAL" creator and REMkill, make visible etc. **FULL** manual with example transfers. **MD1b or WD1b cots on tape £6.99.** "MD1b manages more programs" — Your Spectrum.

To manage the latest programs you will also need our T7 program which INCLUDES the ability to ALTER fast/jerky loaders to "normal".

*Features Baud rate measurer *Jerky detector

T7 cost £6.50 on tape. On M/drive cartridge £9.99 (incl MT6, an extra M/drive utility).

NEW

LERM INFORMATION SHEETS

Each sheet gives you the DETAILED instructions (using our software) required to transfer to your drive 8 of the more POPULAR programs. They cost **£2 EACH plus SAE.** Up to sheet 5 available.

SPECIAL OFFER — TRANSFER PACK. Buy MD1b+T7 on M/Drive cartridge + INFORMATION SHEETS 1-3 for £15.50 (other drive owners get WD1b+T7 on tape + sheets for same price).

UPDATE SERVICE: Send old tape + large SAE to get a £2 REDUCTION. **OVERSEAS:** add £1 Europe, £2 others for each product. **SAE for enquiries.**

LERM, DEPT ZX, 10 BRUNSWICK GDNS, CORBY, NORTHANTS NN18 9ER.

TO ADVERTISE YOUR REPAIRS AND SPARES SERVICE PHONE DAVE 01-437 0699

SPECTRUM REPAIRS
A Better Deal from Micro-World

REPLACEMENT OF MIC, EAR & POWER SOCKET	£8.95
KEYBOARD FAULTS: Replacement of Membrane, Mat & Plate (Note all three are replaced)	£13.95
NEW CASE UPPER & LOWER Inc new keyboard as above	£17.95
ALL OTHER FAULTS 16K Spectrum £17.95 48K Spectrum	£22.95
POWER SUPPLIES (Spectrum – ZX81)	£9.95
16-48K UPGRADES (12 months warranty)	£27.95

- Prices fully inclusive of VAT & Return P&P
- 48hr turn round on most machines
- 3 months warranty on repairs
- Send SPECTRUM ONLY suitably packed clearly stating fault, your name & address, cheque or postal order to:

Micro-World Computers (ZX)
1006/1010 Manchester Rd, Linthwaite, Huddersfield HD7 5QQ
TEL 0484-846117 Showroom open 9-5.30, 6 days

PROFESSIONAL REPAIRS
The Complete Sinclair Service Centre

ZX 81	£13.95
ZX 15k RAM	£13.95
ZX Interface II	£13.95
ZX Interface I	£17.95
ZX Microdrive	£17.95
ZX Spectrum	£17.95
ZX Printer	£17.95
Sinclair QL	P.O.A.
16-48k Upgrade	£26.95

Also BBC, C64, Oric, Dragon, Atari, Amstrad.
RING (0234) 213645
★Average 3-5 day turnaround ★Free-Estimates★All prices inclusive★6 months warranty.

ZEDEM COMPUTER LTD., 2 Kimbolton Road, Bedford.
TRADE ENQUIRIES WELCOME

ZX SPECTRUM REPAIRS
CHECK THESE FACTS
1. FAST RELIABLE SERVICE
2. QUALIFIED STAFF
3. 3 MONTH GUARANTEE
+ + FREE POSTER + +

Send your machine only
Plus a cheque/PO for £19.95

To: MICROTRIX
24 Gt. Moor St. Bolton BL1 1NP
Tel. 0204 398176

BBC SPECTRUM REPAIRS
- ★ Spectrum repairs £18.50 inc. parts insurance and p.p.
- ★ BBC B repairs £18.50 + parts, insurance and p + p.

Send your computer with cheque or P.O. for £18.50 and description of fault to:
SUREDATA (ZX). Computer service
45 Wychwood Avenue, Edgeware, Middx. Tel: 01-951 0124 PL53

COMPUTER REPAIRS
Commodore 64 **£15.00**
Vic 20 **£12.50**
Spectrum **£10.50**
QL **£30.00**
1541 Disk Drives **£20.00**
and all Commodore systems.
Send faulty machine with note describing fault to:
Trident Enterprises Ltd., 37 Linden House, Common Road, Langley, Slough.
TEL: 0753 48755.
Overseas enquiries welcome.
(Above prices are exclusive of parts and VAT)

COMPUTER REPAIRS
We are the experts, having serviced Sinclair computers since the introduction of the ZX80.
Don't waste money on estimates – we repair Sinclair computers at price quoted (inclusive parts, labour, postage, VAT, irrespective of fault. No hidden charges.
Repairs guaranteed for 3 months.

Spectrum	£18.75 inc parts
ZX81	£11.50 inc parts
16 KRam	£9.95 inc parts
Microdrive	£15.95 inc parts
Interface 1-11	£18.75 inc parts
also	
BBC	£22.00 + parts
Electron	£19.95 + parts
XK Memory Expansion Kit	£15.95

Computer Retailers please phone for Special Trade Price.
Call or send with cheque or P.O.
T.V. Services of Cambridge Ltd.
French's Road, Cambridge, CB4 3NP
Tel. 0223 311371

FAST SPECTRUM and BBC REPAIRS
AVERAGE £15

FIRST CO. IN ALL U.K. TO INTRODUCE WHILE-U-WAIT SERVICE. NOW FIRST TO OFFER AN INTERNATIONAL REPAIR SERVICE.

- Most Spectrums repaired in 45 mins
- Please phone for appointment first
- or phone/send Spectrum for free estimates, enc. £1.80 (U.K. I.O.M. Eire, C.I.) £6.50 (Europe/Scandinavia)
- No hidden handling charges
- 24 hr turnaround
- Every micro insured for return journey
- Open till 7pm. Mon - Sat.

"I have come across a firm that will be more than willing to advise you as to how to remedy your problem. They are called **MAN-COMP**, and as well as repairing faulty Spectrums, are also quite willing to discuss your problems with you, and offer reasonably cheap, and (more importantly) correct cures." *Phil Rogers 'Peek & Poke', Popular Computing Weekly, Jan 1985. (Vol IV No. 1)*

MANCOMP LTD. (Dept ZX), Printworks Lane, Manchester, M19 3JP. Tel: 061-224 1888.

SPECTRUM REPAIRS
We expertly repair Spectrums for £17 inclusive of parts and labour and return postage.

Spectrums upgraded to 48K for £34 inc. P&P. All keyboard faults cost only £12.

R. A. Electronics, 133 London Road South, Lowestoft, Suffolk. Tel: (0502) 66289.

SINCLAIR COMPUTER REPAIRS
All machines despatched within 72 hours of receipt.
Spectrum **£17.00** ZX-81 **£12.50**
Spectrum 16 to 48K upgrade **£25.00**
(inc. parts, VAT and return postage)
For a fast and reliable service, send mahcine, brief description of fault and cheque or postal order to the experts:
Capital Computer Services, Sinclair Dept., 44 Rich's Road, Cardiff CF4 4AA. Tel: (0222) 614401.

FAST COMPUTER REPAIRS
VIDEO VAULT INTERNATIONAL 10★ REPAIR SERVICE

Spectrums repaired for £19.95 inc. Parts, Insurance and P & P No Hidden Extras.
BBC, Commodore 64, Vic 20, Atari, Quoted for.
Are you fed up with waiting weeks for your Home Computer to be repaired!!! Why not try the Experts, we offer a full repair service on any Home Computer. All repairs carry a full 3 months guarantee, plus on every Spectrum repaired we send you a free game worth £5.95 for your enjoyment and pleasure.

- ★ **While you wait Service by Professional Computer Engineers.**
- ★ **3 Months written guarantee on all repairs.**
- ★ **International repair company.**
- ★ **All computers sent by mail order turned around in 24 hrs.**
- ★ **Most Spectrums repaired within 45 minutes.**
- ★ **All Micros insured for return journey.**
- ★ **Open 7 days a week./Keyboard faults only £8.95**
- ★ **School repairs undertaken-discounts available.**
- ★ **Free software with each Spectrum repaired worth £5.95.**
- ★ **Over 8 years experience working with computers.**

N.B. WE DO NOT CHARGE AN "AVERAGE PRICE"

ONLY GENUINE SINCLAIR PARTS USED.
Phone today for your free estimate or send your computer to us with £1.75 (UK EIRE I.O.M. C.I.) or £2.40 (Europe/Scandinavia) we will contact you the same day by phone or 1st class Mail should you require quotation first.
TRADE ENQUIRIES MOST WELCOME. ACCESS WELCOMED
VIDEO VAULT LTD THE LOGICAL CHOICE

VideoVault

FREE GAME WORTH £5.95 WITH EACH SPECTRUM REPAIR.

Telephone: Glossop (STD 04574) 66555/67761
140 High St. West, Glossop, Derbyshire, England.

ZX COMPUTING
CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!

Rates of charge: 40p per word per issue plus 15% VAT (minimum of 15 words). Please state classification and post to: **ZX COMPUTING, CLASSIFIED DEPARTMENT, 1 GOLDEN SQUARE, LONDON W1.**

					£5.25
					£7.00
					£8.75
					£10.50
					£12.25
					£14.00
					£15.75
					£17.50

Please place my advert in **ZX COMPUTING** for...... issues commencing as soon as possible.

I am enclosing my Cheque/Postal Order/International Money Order for: (delete as necessary) £...... (Made payable to A.S.P. Ltd)

OR Debit my Access/Barclaycard (Delete as necessary)

VISA

We welcome Access

All classified advertisements must be paid for in advance.

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)

Address ....

....

....

Signature .... **Date** ....

Daytime Tel. No. ....

Cut out and SEND TO:

ZX COMPUTING

ZX Computing Subscriptions,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB,

SUBSCRIPTION RATES

(tick ☐ as appropriate)

£13.50 for six issues	☐
£15.00 for six issues	☐
£31.00 for six issues	☐
$19.50 for six issues USA surface mail.	☐

Subscription Order Form

Please commence my subscription to ZX Computing with the very next issue.

I am enclosing my (delete as necessary) cheque/Postal Order/International Money Order for £:..........
(made payable to ASP Ltd)
OR
*Debit my Access/Barclaycard**
*(*delete as necessary)*

We welcome Access

VISA

Please use BLOCK CAPITALS and include post codes.

NAME (Mr/Mrs/Miss) ....

ADDRESS ....

....

.... **POSTCODE** ....

Signature ....

Date ....


GLASS
Battle your way to three enemy bases.
Do you have the courage, stamina
and skill to become a Game Lord?
£7.99
SPECTRUM 48K
From
QUICKSILVA
The most powerful games in the Universe!
Available from all good software shops.
In case of difficulty write to:
QUICKSILVA Ltd. Liberty House, 222 Regent Street, London W1R 7DB tel: 01 439 0666
Argus Press Software Group


Nick Faldo's
Open
£9.99
Runs on both Spectrum 48K and CBM64
Accurate 900 screen scrolling map of the Royal St. George's Golf Course, venue for the Open Championship 1985.
Can you make par on this course of Champions? With strong winds and deep bunkers you'll need all the help your caddy can give you. Full animation and live action bring your game to life.
MIND GAMES
Argus Press Software Ltd., Liberty House, 222 Regent Street, London W1R 7DB Telephone 01-439 0666
Free booklet featuring maps and history of the course with a hole by hole guide by